Informatica

Informatica® Test Data Management 10.2 R1

User Guide

Informatica Test Data Management User Guide
10.2 R1
June 2018

© Copyright Informatica LLC 2003, 2018

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

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

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.

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

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

Publication Date: 2018-07-19

Table of Contents

Preface 18

Informatica Resources. 18

Informatica Network. 18

Informatica Knowledge Base. 18

Informatica Documentation. 18

Informatica Product Availability Matrixes. 19

Informatica Velocity. 19

Informatica Marketplace. 19

Informatica Global Customer Support. 19

Chapter 1: Introduction to Test Data Management. 20

Test Data Management Overview. 20

Test Data Management Use Cases. 22

Using TDM to Improve Performance in Application Testing Environments. 22

Using TDM for Security and Compliance in Application Testing Environments. 22

Using TDM With Data Validation Option to Create Verified Test Data. 23

TDM Architecture. 24

TDM Tools. 26

TDM Server. 26

Application Services. 26

TDM Databases. 27

TDM Connections. 28

TDM Process. 30

Create a Data Masking Policy. 30

Create a Project and Import Metadata. 30

Discover Source Information. 31

Define Data Masking and Data Subset Operations. 31

Define a Data Generation Operation. 31

Create a Plan for Data Masking and Data Subset. 31

Create a Plan for Data Generation. 32

Monitor the Workflow. 32

TDM Example. 33

Using Advanced Parameterization in Test Data Manager. 34

Chapter 2: Test Data Manager. 36

Test Data Manager Overview. 36

Test Data Manager User Interface. 37

Views. 37

Search Field. 38

Advanced Search Text Filter. 38

Quick Links Menu. . . . 39
User Preferences Menu. . . . 39
Actions Menu. . . . 39
Data Filter. . . . 39
Overview View. . . . 40
Dashboards. . . . 40
Project Risk Analysis. . . . 40
Project Assignments and Project Objects. . . . 41
Recent Project Sensitive Data Distribution. . . . 41
Recent Activities. . . . 41
Plan Execution. . . . 41
Global Components. . . . 41
Recent Projects. . . . 41
Policies View. . . . 42
Data Sets View. . . . 42
Projects View. . . . 43
Monitor View. . . . 44
Administrator View. . . . 44
Expression Builder. . . . 45
Logging In to Test Data Manager. . . . 46

Chapter 3: Projects. . . . 48

Projects Overview. . . . 48
Project Components. . . . 49
Parameters. . . . 50
Rules and Guidelines for Parameters. . . . 51
Project Logs. . . . 51
Data Masking Transformation Session Properties. . . . 51
Project Management. . . . 52
Creating a Project. . . . 52
Editing a Project. . . . 53
Copying a Project. . . . 54
Deleting a Project. . . . 54
Creating a Parameter. . . . 54
Editing or Deleting a Parameter. . . . 55
Configuring Project Log Locations. . . . 55
Exporting a Project. . . . 56
Importing a Project. . . . 56
Data Sources. . . . 56
TDM Source Definitions. . . . 57
Hadoop HDFS Data Sources. . . . 57
XSD Data Sources. . . . 57
Editing the Properties of an Element in an XSD File. . . . 59

XSD Import Guidelines. . . . 60
PowerExchange Nonrelational Sources. . . . 60
Importing Data Sources. . . . 60
Identity Columns in TDM. . . . 61
Delete a Table. . . . 62
Deleting a Table. . . . 62
Project Permission and Security. . . . 62
Project Permissions. . . . 63
Updating User and Group Security. . . . 63

Chapter 4: Policies. . . . 65

Policies Overview. . . . 65
Policies View. . . . 66
Policies Task Flow. . . . 66
Rules. . . . 66
Data Domains. . . . 67
Apply Masking Rules to a Data Domain. . . . 67
Apply Generation Rules to a Data Domain. . . . 67
Metadata and Data Patterns for Data Domains. . . . 68
Regular Expression Syntax. . . . 68
Data Patterns 69
Metadata Patterns. . . . 69
Data Domain Options. . . . 69
Creating a Data Domain. . . . 70
Copying a Data Domain. . . . 70
Editing a Data Domain. . . . 70
Deleting a Data Domain. . . . 71
Policy Packs. . . . 71
PII Policy Pack. . . . 72
PHI Policy Pack. . . . 73
PCI Policy Pack. . . . 74
Import and Export. . . . 74
Exporting Policy Components. . . . 74
Importing Policy Components. . . . 75
Data Domain Import. . . . 75
Linking Business Glossary Terms to Global Objects 76
Linking a Business Term to an Object. . . . 76
Deleting a Business Term Link to an Object. . . . 77
Policy Management. . . . 77
Creating a Policy. . . . 77
Copying a Policy. . . . 78
Editing a Policy. . . . 78
Deleting a Policy. . . . 78

Chapter 5: Data Discovery 80

Data Discovery Overview 80
Data Discovery Sources 81
Rules and Guidelines for Data Discovery Sources 81
Discover View 82
Column Properties 83
Sensitive Columns 83
Computed Columns 83
Value Cascades 84
Auto Cascades 85
Data Discovery Task Flow 85
Primary Key Discovery 86
Primary Key Profile Options 86
Entity Discovery 87
Entity Profile Options 87
Data Domain Discovery 88
Data Domain Profile Sampling Options 88
Assigning a Data Domain to Multiple Columns 89
Manually Updating the Column Data Domain 89
Data Domain Assignments Import and Export 89
Column Profile 90
Inferred Rules 90
Column Profile Options 91
Profile Management 91
Creating a Primary Key Profile 92
Creating an Entity Profile 92
Creating a Data Domain Profile 93
Creating a Column Profile 94
Editing a Profile 94
Deleting a Profile 95
Profile Import 95
Importing a Profile 96
Apply the Results 96
Primary Key Results 97
Entity Discovery Results 98
Data Domain Discovery Results 98
Column Profile Results 99
Project Tables 100
Table Classification 101
Constraints 101
Manually Add Keys to Tables in a Project 102

Creating a Primary Key Constraint. 103
Creating a Logical Relationship Between Tables. 103
Creating a Unique Key Constraint. 104
Creating a Conditional Constraint. 104

Chapter 6: Creating a Data Subset. 106

Data Subset Overview. 106
Data Subset Process Flow. 107
Data Subset Components. 107
Entities. 108
Groups. 110
Creating an Entity. 110
Entity Constraint Properties. 111
Parameters in Entity Criteria. 111
Database Functions in Entity Criteria 112
Optimizing Relations. 113
Disabling and Enabling Relationships in an Entity. 114
Review Changes. 114
Entity Example. 115
Creating a Group. 115
Applying Criteria to an Element or Attribute. 116
Editing a Data Subset Component. 116
Exporting a Data Subset Component. 117
Importing a Data Subset Component. 117
Copying a Data Subset Component. 118
Deleting a Data Subset Component. 118
Creating a Data Subset. 118
Example - Data Subset for XSD Data Sources. 119
Example XML Structure Definition. 120
Example Master XML Data. 121
Data Subset for Group A. 124
Data Subset for Group B. 125

Chapter 7: Performing a Data Masking Operation. 127

Data Masking Overview. 127
Data Masking Task Flow. 128
Data Masking Rules. 128
Standard Masking Rules 129
Mapplet Rules. 129
Advanced Masking Rules. 129
Masking Rule Assignments. 130
Creating and Assigning Data Masking Rules. 131
Creating a Standard Masking Rule. 131

Creating a Mapplet Masking Rule. . . . 132
Creating an Advanced Masking Rule. . . . 133
Adding Data Masking Rules to a Project. . . . 133
Assigning a Standard Masking Rule. . . . 134
Assigning a Custom Masking Rule. . . . 134
Assigning an Advanced Masking Rule. . . . 135
Applying a Data Masking Rule to XML Elements and Attributes. . . . 135
Modifying Data Masking Rules and Assignments. . . . 136
Available Data Masking Rules 137
Editing a Masking Rule. . . . 137
Copying a Masking Rule. . . . 138
Deleting a Masking Rule. . . . 138
Overriding a Masking Rule. . . . 138
Assigning Single and Multiple Masking Rules. . . . 139
Deleting Masking Rule Assignments. . . . 140
Performing a Data Masking Operation. . . . 140
Data Masking Components. . . . 141
Masking Components in PowerCenter. . . . 142

Chapter 8: Data Masking Techniques and Parameters. . . . 143

Data Masking Techniques and Parameters Overview. . . . 144
Data Masking Techniques. . . . 144
Data Masking Parameters. . . . 145
Repeatable Output. . . . 146
Exception Handling. . . . 147
Custom Masking. . . . 148
Custom Masking Parameters. . . . 148
Advanced Masking. . . . 148
Advanced Masking Parameters. . . . 149
Advanced Masking Example. . . . 150
Credit Card Masking. . . . 150
Credit Card Masking Parameters. . . . 151
Email Masking. . . . 151
Email Masking Parameters. . . . 151
Encryption Masking. . . . 153
Encryption Masking Parameters. . . . 154
Expression Masking. . . . 155
Expression Masking Parameters. . . . 155
Rules and Guidelines for Expression Masking. . . . 155
IP Address Masking. . . . 156
Key Masking. . . . 156
Mask Format. . . . 156
Source String Characters. . . . 157

Result String Replacement Characters. . . . 157
Date Key Masking . . . 158
Numeric Key Masking Parameters. . . . 158
String Key Masking Parameters. . . . 158
Nullification Masking. . . . 159
Phone Masking. . . . 159
Random Masking. . . . 159
Range Masking. . . . 160
Blurring. . . . 160
Mask Format. . . . 161
Source String Characters. . . . 161
Result String Replacement Characters. . . . 162
Date Random Masking Parameters. . . . 162
Numeric Random Masking Parameters. . . . 163
String Random Masking Parameters. . . . 163
Shuffle Masking. . . . 164
Shuffle Masking Parameters. . . . 164
SIN Masking. . . . 165
SSN Masking. . . . 165
Substitution Masking. . . . 166
Substitution Masking Parameters. . . . 167
URL Masking. . . . 167
Name Substitution Example . . . 168
Add a Dictionary in Test Data Manager. . . . 168
Creating the Substitution Rule. . . . 169
Creating the Advanced Masking Rule. . . . 170
Shuffle Address Example. . . . 174
Creating the Shuffle Rule. . . . 174
Create the Advanced Masking Rule. . . . 175

Chapter 9: Data Generation 181

Data Generation Overview. . . . 181
Data Generation Components. . . . 182
Data Generation Task Flow. . . . 183
Data Generation Rule Types. . . . 183
Default Settings. . . . 184
Define Default Generation Rules. . . . 185
Standard Generation Rules. . . . 185
Creating a Standard Generation Rule. . . . 186
Editing a Generation Rule. . . . 186
Overriding a Generation Rule. . . . 186
Deleting a Generation Rule. . . . 187
Custom Generation Rules. . . . 187

Creating a Custom Generation Rule. . . . 187
Ad Hoc Generation Rules. . . . 188
Creating an Ad Hoc Generation Rule. . . . 188
Editing an Ad Hoc Generation Rule. . . . 189
Advanced Generation Rules. . . . 190
Advanced Generation Rule Example. . . . 190
Creating an Advanced Generation Rule. . . . 191
Conditional Constraints. . . . 192
Conditional Constraints and Data Conflicts. . . . 193
Data Generation Rule Assignments. . . . 193
Available Data Generation Rules. . . . 194
Assigning Standard Generation Rules to Columns. . . . 194
Assigning Custom Generation Rules to Columns. . . . 195
Assigning Advanced Generation Rules to Columns. . . . 195
Assigning Auto Rules to Columns. . . . 196
Deleting Generation Rule Assignments. . . . 196
Data Generation for XSD Sources. . . . 197
Data Generation Task Flow for XSD Sources. . . . 197
Applying Data Generation Rules to XML Elements and Attributes. . . . 197
Data Generation Plans and Workflows. . . . 198

Chapter 10: Data Generation Techniques and Parameters. . . . 199

Data Generation Techniques and Parameters Overview. . . . 199
Data Generation Techniques. . . . 200
Data Generation Parameters. . . . 201
Exception Test Data. . . . 201
Custom Generation. . . . 202
Custom Generation Parameters. . . . 203
Advanced Generation. . . . 203
Advanced Generation Parameters. . . . 205
Dictionary Generation. . . . 206
Dictionary Generation Parameters. . . . 207
Effective Dates Generation. . . . 208
Effective Dates Generation Example. . . . 208
Effective Dates Generation Parameters. . . . 209
Expression Generation. . . . 210
Expression Generation Parameters. . . . 211
Random Generation. . . . 211
Date Random Generation Parameters. . . . 212
String Random Generation Parameters. . . . 213
Numeric Random Generation Parameters. . . . 214
Data Patterns for Random Generation. . . . 214
Credit Card Number Generation. . . . 217

Issuer Identification Number. . . . 218
Credit Card Generation Parameters. . . . 219
Reference Lookup Generation. . . . 220
Reference Lookup Generation Parameters. . . . 221
Sequence Generation. . . . 221
Date Sequence Generation Parameters. . . . 222
Numeric Sequence Generation Parameters. . . . 223
Set of Values Generation. . . . 223
Set of Values Generation Parameters. . . . 224
Conditional Generation. . . . 225
Conditional Generation Parameters. . . . 226

Chapter 11: Working with Test Data Warehouse. . . . 227

Test Data Warehouse Overview. . . . 227
Test Data Warehouse Process. . . . 228
Data Sets. . . . 229
Data Set Tags. . . . 230
Test Data Management Self-Service Portal. . . . 230
Creating a Data Set. . . . 230
Reset a Data Set. . . . 231
Classification of Data Set Tables. . . . 232
Classifying a Data Set Table. . . . 232
Reset With and Without Downtime. . . . 233
Resetting a Data Set. . . . 234
Reset Process Flow. . . . 235
Reset with Truncate Tables Process Flow. . . . 236
Reset Without Truncate Tables Process Flow. . . . 237
Restoring Deleted Data After a Failed Reset. . . . 238
Editing the Metadata of a Data Set. . . . 238
Publishing a Data Set to the Self-Service Portal. . . . 238
Deleting a Data Set. . . . 239
Related Data Sets. . . . 239
Related Data Set Lineage. . . . 240
Creating an Entity. . . . 240
Editing an Entity. . . . 241
Creating a Group. . . . 241
Editing a Group. . . . 242
Creating a Related Data Set. . . . 242
Copying a Data Set Plan. . . . 243
Data Set Permissions. . . . 243
Editing Data Set Permission. . . . 244
Locking and Unlocking a Data Set. . . . 245
Monitor a Data Set Job. . . . 245

View and Manage Data in a Data Set. 246
Rules and Guidelines for Viewing and Managing Data. 246
Viewing Data in a Data Set. 246
Using the SQL Query Field to View and Manage Data. 247
SQL Queries to View and Edit Data. 247
Updating Data in the Grid. 248
Adding Rows to a Data Set Table. 248
Deleting Rows in a Data Set Table. 248
Managing Row-level Tags in a Data Set Table. 249

Chapter 12: Analyzing Test Data with Data Coverage. 250

Data Coverage Analysis Overview. 250
Data Coverage Process. 251
Creating a Data Coverage Task. 251
Data Coverage Task Columns. 252
Data Type Exceptions. 254
Creating a Data Coverage Column Using Data As Is. 255
Creating a Data Coverage Column Using Data Ranges. 255
Creating a Data Coverage Column Using Mappings. 256
Data Coverage Analysis Page. 257
Editing a Data Coverage Task. 258
Marking a Cell as Invalid. 258
Updating Data Across Cells. 259
User Input in Fill Cell Jobs. 260
User Input Example. 260
Example Data. 260
Data Coverage Analysis. 261
Data Coverage Analysis Example. 262
Tables in the Data Set. 262
Analysis for Data Coverage. 263

Chapter 13: Plans and Workflows. 265

Plans and Workflows Overview. 265
Plans and Workflows Task List. 266
Workflow Connections. 267
Plan Components. 267
Pre Workflow and Post Workflow Parameters. 268
Plan Settings. 268
Connection Options. 269
Data Set Properties. 270
Test Tool Integration Options. 271
File Path Settings. 271
Target Options. 272

Update Strategy Options. . . . 273
Data Generation Properties. . . . 273
Error and Recovery Options. . . . 274
Encryption Options. . . . 275
Advanced Options. . . . 275
Log Settings. . . . 279
Masking Components. . . . 280
Subset Components. . . . 280
Generation Components. . . . 281
Component Criteria. . . . 281
Filtering Data Subset Components. . . . 281
Disabling Masking for a Column. . . . 282
Source Settings. . . . 282
Connection Properties. . . . 282
Test Tool Integration Properties. . . . 285
Target Properties. . . . 285
Update Strategy Properties. . . . 285
Error and Recovery Properties. . . . 286
Source and Target Properties. . . . 286
Advanced Properties. . . . 287
Partitioning Properties. . . . 288
XSD Data Source Connection Settings. . . . 288
XSD Data Source Advanced Settings. . . . 289
Plan Management. . . . 290
Creating a Data Masking and Data Subset Plan. . . . 290
Creating a Data Generation Plan. . . . 291
Creating an Ad Hoc Generation Plan. . . . 291
Copying a Plan. . . . 292
Exporting a Plan. . . . 292
Importing a Plan. . . . 292
Deleting a Plan. . . . 293
Workflow Generation. . . . 293
Generating a Workflow. . . . 293
Export to HP ALM. . . . 293
Workflow View. . . . 294
Parameter Files in Test Data Manager. . . . 294
Creating a Parameter File. . . . 295
Executing a Workflow. . . . 295
Workflow Executions View. . . . 296
Workflow Tasks. . . . 297
Workflow Properties Panel. . . . 298

Chapter 14: Monitor 300

Monitor Overview 300
Jobs 301
Job Details 302
Monitor Tasks 302
Logs 303
Severity Levels 303
Viewing the Log Messages 304
Sessions 304

Chapter 15: Reports 306

Reports Overview 306
Audit Trail Report 307
Running an Audit Trail Report 307
Data Masking Report 307
Running the Data Masking Report 307
Plan Audit Report 307
Running a Plan Audit Report 308
Plan Detail Report 309
Running the Plan Detail Report 309
Row Count Report 309
Running the Row Count Report 310

Chapter 16: ilmcmd 311

ilmcmd Overview 311
Configuring ilmcmd 312
Running ilmcmd 312
Entering Options and Arguments 313
Syntax Notation 313
Delete 314
Delete Examples 314
Export 315
Export Examples 316
Import 316
Import Examples 317
Search 317
Search Examples 318
Workflow 319
Workflow Examples 320
Reset 320
ListPlans 322
TDWPlanGenerate 323

TDWPlanExecute. 324
TDWPlanGenExe. 324

Chapter 17: tdwcmd. 326

tdwcmd Overview. 326
Running tdwcmd. 326
Entering Options and Arguments. 327
Syntax Notation. 327
List. 328
List Examples. 330

Chapter 18: tdwquery. 331

tdwquery Overview. 331
Configuring tdwquery. 331
Running tdwquery. 332
Select Clause. 333
Joins. 333
Aggregate Functions. 334
Where Clause. 335
Additional Clauses. 337

Appendix A: Data Type Reference. 338

Data Type Reference Overview. 338
Oracle. 339
Microsoft SQL Server. 343
DB2 for Linux, UNIX, and Windows. 347
Sybase ASE. 349
Hadoop HDFS. 352
MySQL. 353
Flat File. 354
VSAM Flat. 355
DB2 for z/OS. 356
DB2 for IOS. 358
IMS Flat. 360
Sybase IQ. 361
Netezza. 363
Teradata. 365
Cassandra. 367
MongoDB. 368

Appendix B: Data Type Reference for Test Data Warehouse. 370

Data Type Reference for Test Data Warehouse Overview. 370
Oracle. 371

Microsoft SQL Server. 373
DB2 for Linux, UNIX, and Windows. 375
DB2 for z/OS. 376
Sybase ASE. 377
Teradata. 379
MongoDB. 381

Appendix C: Glossary. 382

Index. 389

Preface

The Informatica *Test Data Management User Guide* describes how to protect sensitive data and create lean non-production systems for test and development. It shows you how to implement data subset, data masking, data generation, and data discovery operations. This guide is written for users who use Test Data Manager with PowerCenter. It assumes knowledge of operating systems, database engines, flat files, and PowerCenter®.

Informatica Resources

Informatica Network

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

As a member, you can:

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

Informatica Knowledge Base

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

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

Informatica Documentation

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

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

Informatica Product Availability Matrixes

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

Informatica Velocity

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

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

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

Informatica Marketplace

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

Informatica Global Customer Support

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

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
http://www.informatica.com/us/services-and-training/support-services/global-support-centers.

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

CHAPTER 1

Introduction to Test Data Management

This chapter includes the following topics:

- Test Data Management Overview, 20
- Test Data Management Use Cases, 22
- TDM Architecture, 24
- TDM Process, 30
- TDM Example, 33
- Using Advanced Parameterization in Test Data Manager, 34

Test Data Management Overview

Test Data Management (TDM) integrates with PowerCenter, PowerExchange®, and Informatica applications to manage nonproduction data in an organization.

With TDM, an organization can create a smaller copy of the production data and mask the sensitive data. An organization can discover the sensitive columns in the test data, and ensure that the sensitive columns are masked in the test data. An organization can also create test data that does not contain sensitive data from the production database. They can create a test data warehouse to store test data in a central location and edit or reset the data when required.

Organizations create multiple copies of application data to use for testing and development. Organizations often maintain strict controls on production systems, but data security in nonproduction systems is not as secure. An organization must maintain knowledge of the sensitive columns in the production data and ensure that sensitive data does not appear in the test environment. Development must not have to rewrite code to create test data.

Manage data discovery, data subset, data masking, and data generation in Test Data Manager.

Data discovery

Use data discovery to run sensitive field profiles to identify the columns that contain sensitive data. Use the profile results to determine which columns to mask and which data masking techniques to apply. Define data domains to identify sensitive data columns by patterns in the data or the column metadata. When you apply data masking, you can apply the same rule to multiple columns in the same data domain. You can run primary and foreign key profiles to discover potential primary key-foreign key constraints to define relationships between parent and child tables.

Data subset

Use data subset to create a small environment for testing and development. You can define the type of data that you want to include in the subset database. You might create a subset database with data based on time, function, or geographic location. For example, a time-based subset database might include recent payment transactions from all invoice data in a production system.

Data masking

Create data masking rules to apply to source columns and data domains. You can apply different masking techniques such as substitution masking, shuffle masking, key masking, and encryption. You can configure repeatable results in the masked data. You can assign multiple rules to the same column.

Data generation

Use data generation to create a testing environment that does not use data from the production database. Create data generation rules to define the type of data you want to generate. TDM generates data in a schema that you can use for testing.

Manage Data Sets

Manage data sets that you store in the test data warehouse. You can view and edit the data or reset the data to a test system from Test Data Manager. For example, you run multiple test cases, or multiple test teams work on an application. You can store the test data in the test data warehouse. When one test team completes testing, save the modified test data as another version of the original data set in the test data warehouse. Restore the required version from the test data warehouse to the test environment to run other test cases or for a different team to work with.

Data coverage analysis

Create data coverage tasks to analyze the data in a data set. You can visualize the data coverage across pairs of columns and use filter columns to further configure the analysis. Based on the results, you can choose to update data to ensure that you have enough data in required cells to meet the test case requirements.

To perform data subset and masking operations, you can generate and run workflows from data subset and data masking plans in Test Data Manager.

To perform data generation operations, you can generate and run data generation plans in Test Data Manager.

You can export test data to an HP ALM server from TDM. You can copy results of subset, masking, and generation operations that have flat file targets. Integrate the HP-ALM test tool with TDM to directly copy and maintain flat file results in an HP ALM server. You can then use the data to create and run test cases in HP ALM.

Configure a test data warehouse to create and work with data sets in Test Data Manager. Create a data set from any data subset, data masking, or data generation plan that you run in Test Data Manager.

TDM includes the ilmcmd command line program. Run ilmcmd commands to perform a subset of the Test Data Manager tasks from the command line.

TDM users have roles and privileges that determine the tasks that they can perform through Test Data Manager or the ilmcmd command line program. The administrator manages roles and privileges for users from the Informatica Administrator.

Test Data Management Use Cases

Uses cases for Test Data Management include improving performance, security, and compliance in application testing environments.

Using TDM to Improve Performance in Application Testing Environments

Testing teams need to be efficient and agile to create and maintain test data while maintaining a collaborative testing environment. You can use the features available in TDM to help you meet this requirement.

You can perform the following tasks to improve performance in a testing environment:

- You can create subsets of production data to use as test data. The data subset retains primary and foreign key relations and the data remains referentially intact. Use data subset to create manageable and scalable data for testing.
- You can directly copy flat file test data from TDM to an integrated test tool and use the data to run test cases.
- You can generate data based on rules that define the type of data required. Use data generation when you need test data but do not want to use production data. For example, consider test teams that work on applications that handle sensitive data. The test teams do not have access to production data. They can use data generation to create test data.
- You can store multiple versions of test data in a common location. Use the test data warehouse as a single and secure location to maintain required versions of test data. You can store data from multiple machines that host a Test Data Manager Service in a domain in a single test data warehouse. Multiple testing teams can then share the stored data and collaborate efficiently.
- You can edit the data in a data set that you create. Edit the data, apply row-level tags to the data, to track and update test data from Test Data Manager.
- You can restore a data set version to return the test environment to a particular state. For example, you might need to return the test environment to an older state or to the original state after you run some test cases. Use the test data warehouse to reduce the time and effort to create and manually manipulate test data.

Using TDM for Security and Compliance in Application Testing Environments

Testing teams need to be compliant and maintain strict controls on the data used in testing environments.

You must identify sensitive information in huge volumes of production data and then ensure that you do not expose the information in the test data. This is a challenge and a compliance and security risk.

You can perform the following tasks to help with security and compliance in a test environment:

- To analyze the data and identify sensitive information in the data, run profiles in TDM. You can compare the data against multiple compliance regulation standards to ensure that the data meets standard compliance regulations. For example, run a profile on the data to identify columns that contain sensitive data according to PII, PCI, or PHI compliance regulations.
- To remove sensitive data that you identify, run a data masking operation in TDM. You can use standard masking formats such as credit card numbers or Social Security numbers to mask data or create customized formats to mask data.

- To reduce the risk associated with creating multiple copies of test data, store the data in the test data warehouse. Multiple teams that use different machines that host a Test Data Manager Service in a domain can access a single test data warehouse.

Using TDM With Data Validation Option to Create Verified Test Data

Data Validation Option (DVO) is a solution that you use with PowerCenter to validate data. You can validate target data to verify that it is accurate and that the transformation process did not introduce errors or inconsistencies.

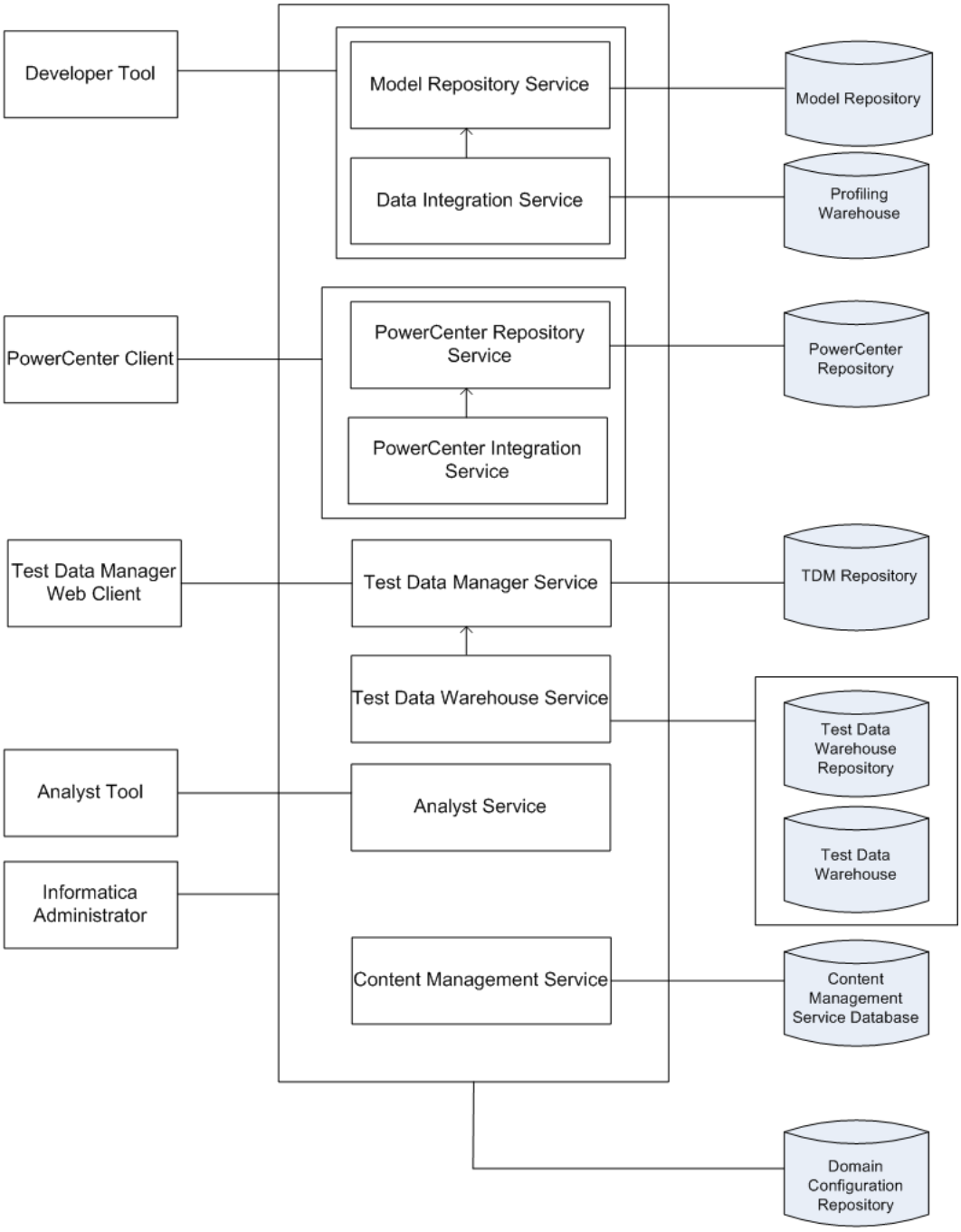
Perform data validation to compare production and development environments when you use TDM to create test data.

For example, consider test teams that need to submit compliance reports for test data created from production data. You can use TDM with DVO to deliver test data with verifiable proof that the test data does not contain sensitive data values.

TDM Architecture

The TDM architecture consists of tools, the Test Data Manager Service and other application services, and databases.

The following image shows the components of TDM:



The following table describes the architecture components:

Component	Description
Test Data Manager	A web-based client that you can use to perform data discovery, data subset, data masking, data generation operations, and operations on data sets in the test data warehouse.
Developer Tool	A thick client that you use to create and run profiles to analyze the data.
PowerCenter Client	The PowerCenter Client application consists of the tools to manage the repository and to design mappings, mapplets, and sessions to load the data.
Informatica Administrator	A web application that you can use to manage, monitor, deploy, and undeploy data flows.
Model Repository Service	An application service that manages the Model repository.
Data Integration Service	An application service that performs data integration tasks for the Developer tool and external clients and data coverage tasks for the test data warehouse.
Test Data Manager Service	An application service that runs Test Data Manager and manages connections between service components and Test Data Manager users.
PowerCenter Repository Service	An application service that manages the PowerCenter repository where the metadata for workflows and mappings that you generate in Test Data Manager are stored.
PowerCenter Integration Service	An application service that runs sessions and workflows that you run from Test Data Manager.
Test Data Warehouse Service	An application service that runs the test data warehouse and manages the test data warehouse repository and the test data warehouse.
Content Management Service	An application service that manages reference data.
Analyst Service	An application service that manages the connections between service components and the users that have access to the Business Glossary.
Service Manager	The Informatica Service that manages all domain operations and manages the domain configuration repository.
TDM repository	A relational database that stores the components that you define in Test Data Manager, such as policies, projects, entities, and rules. The TDM repository stores metadata that you import into Test Data Manager from a source database or from the PowerCenter repository.
Test data warehouse repository	A relational database that stores the source table metadata and project metadata for data sets that you create.
Test data warehouse	A relational database that stores the source data that you include in a data set.
Profiling warehouse	A relational database that stores profile results.
Model repository	A relational database that stores the table metadata for data discovery profiles. The Model repository also stores connection information for connections that you create in TDM.

Component	Description
Domain configuration repository	A relational database that stores the connections used to run profiles, users for the Informatica domain, and metadata for the Informatica domain.
PowerCenter repository	A relational database that stores the metadata for workflows and mappings that you generate from Test Data Manager. It stores information required to extract, transform, and load data. It also stores administrative information such as permissions and privileges for users and groups that have access to the repository.

TDM Tools

The TDM tools consist of Test Data Manager, Informatica Developer, Informatica Administrator, and the PowerCenter Client.

Test Data Manager

A web-based client application that you can use to configure data masking, data subset, data generation, and profiles for data discovery. You can also configure connections, and manage project permissions for users and user groups.

Informatica Developer

A client application that you use to create and export profiles for data discovery.

Informatica Administrator

A web-based client that a domain administrator uses to manage application services and create users and user groups.

PowerCenter Client

A client application that you use to configure permissions on folders and on connection objects for the PowerCenter repository.

TDM Server

The TDM server is the interface between Test Data Manager and the application services.

Application Services

TDM uses Informatica services. Create the services in the Administrator tool.

TDM uses the following application services:

PowerCenter Repository Service

An application service that manages the PowerCenter repository. The PowerCenter Repository Service accepts requests from the PowerCenter Integration Service when a workflow runs.

PowerCenter Integration Service

An application service that runs data subset, data generation, and data masking workflows. When you create the Test Data Manager Service in the Informatica Administrator, you select the PowerCenter Integration Service that runs the workflows.

Model Repository Service

An application service that manages the Model repository for data discovery operations.

Data Integration Service

An application service that performs the data discovery operations. The Data Integration Service connects to the Model Repository Service to store metadata from data discovery profiles in the Model repository. When you create a Data Integration Service in the Administrator tool, you select the data profiling warehouse to store data from data discovery profiles. The Data Integration Service performs data coverage tasks for the test data warehouse.

Test Data Warehouse Service

An application service that manages the test data warehouse repository and the test data warehouse. The Test Data Warehouse Service connects to the test data warehouse repository to copy the metadata created when you create a data set. It creates the tables for data sets in the test data warehouse.

Analyst Service

Manages the Analyst tool that TDM uses to link TDM global objects to objects in the Business Glossary. The license for the Analyst Service to use database content from the TDM repository.

Test Data Manager Service

An application service that creates and manages the TDM repository. The Test Data Manager accesses the Test Data Manager Service to use database content from the TDM repository.

TDM Databases

TDM connects to databases for PowerCenter metadata, profiling, TDM configuration, domain configuration, and the test data warehouse.

TDM needs connections to the following databases:

TDM repository

A relational database that contains tables that TDM requires to run and the tables that store metadata.

Model repository

A relational database that stores table metadata for data discovery profiles and the connections that you create in Test Data Manager.

PowerCenter repository

A relational database that stores metadata for PowerCenter sources and targets. The PowerCenter repository also stores metadata for the workflows that you generate from plans in Test Data Manager.

Profiling warehouse

A relational database that stores profile results for data discovery.

Domain configuration repository

A relational database that stores connections and metadata for the Informatica domain.

Test data warehouse repository

A relational database that stores the project metadata and source table metadata for data sets that you store in the test data warehouse.

Test data warehouse

A relational database that stores the source data of data sets that you store in the test data warehouse.

TDM Connections

TDM connects to databases, repositories, and services to perform data subset, masking, generation, and profiles for discovery operations. The connection requirements are based on the operations that you need to perform.

To perform data discovery operations, TDM requires connections to a database source and a Data Integration Service.

Workflows for data subset and masking operations require connections to services, the TDM repository, external database sources, and PowerCenter sources and targets.

Workflows for operations to store data in the test data warehouse require connections to external database sources, the test data warehouse repository, and the test data warehouse.

To run workflows to copy flat file results to a test tool, you must first integrate the test tool server with TDM. TDM connects to the integrated server to copy the flat file to the configured location.

Service Connections

TDM requires connections to the following services:

Data Integration Service

TDM requires a connection to a Data Integration Service for data discovery operations. The Data Integration Service is the service in the Informatica domain that performs the data discovery operation.

Model Repository Service

An application service that manages the Model repository for data discovery operations.

PowerCenter Integration Service

An application service that runs workflows that users generate in Test Data Manager for TDM operations. A data masking, data generation, or data subset workflow requires a connection to a PowerCenter Integration Service. When you generate a workflow from a plan, select the name of the PowerCenter Integration Service to use.

PowerCenter Repository Service

An application service that manages the PowerCenter repository. When you import metadata into Test Data Manager from a PowerCenter source, the TDM Server sends a request to the PowerCenter Repository Service to extract source metadata from the PowerCenter repository and load it to the TDM repository.

Analyst Service

An application service that manages the Analyst tool. TDM uses the Analyst tool to link TDM global objects to objects in the Business Glossary. The license for the Analyst Service must support Business Glossary.

Test Data Warehouse Service

An application service that manages the test data warehouse repository and the test data warehouse. The Test Data Warehouse Service connects to the test data warehouse repository to copy the metadata created when you create a data set. It creates the tables for data sets in the test data warehouse.

Test Data Manager Service

The TDM application service that manages the TDM repository. Test Data Manager accesses the Test Data Manager Service to use database content from the TDM repository and to connect to other services to perform TDM operations.

Repository Connections

TDM requires connections to repositories.

TDM accesses the following repositories:

TDM repository

The PowerCenter Integration Service stores TDM components in the TDM repository.

A data masking, data generation, or data subset workflow requires a connection to the TDM repository.

Model repository

When you run profiles to discover data, the TDM Server sends a request to the Data Integration Service to extract data for the source tables. The Data Integration Service sends a request to its associated Model Repository Service to load the metadata for the tables to the Model repository.

Profiling warehouse

The Data Integration Service loads the profile results to the profiling warehouse. When you create a Data Integration Service in the Administrator tool, you configure a profiling warehouse.

PowerCenter repository

When you import metadata into Test Data Manager from a PowerCenter source, the TDM Server sends a request to the PowerCenter Repository Service to copy source metadata from the PowerCenter repository into the TDM repository.

Test data warehouse repository

When you run a TDM operation to create a data set in the test data warehouse, the Test Data Manager Service stores the project metadata and source table metadata in the test data warehouse repository.

Test data warehouse

A relational database that stores the source data that you include in data sets that you store in the test data warehouse.

Database Connections

TDM requires a connection to a database source to perform data discovery operations and a database target to perform data generation operations. It also requires connections for relational sources and targets for data subset and data masking. To store data sets in the test data warehouse, TDM requires a connection to a source and to a test data warehouse.

Configure connections in the Administrator view of Test Data Manager. To connect TDM to databases, you do not need to install a separate driver.

When you generate workflows from data subset or masking plans, you must select connections for relational sources and targets. When you generate workflows from data generation plans, you must select connections for targets. Select these connections in Test Data Manager when you create a plan. TDM tests the connections when it generates workflows from plans, and the PowerCenter Integration Service uses the connections when it runs the workflows.

TDM Process

Run a profile against source data, create a subset of the data, and mask the subset data.

The TDM process includes the following high-level steps:

1. Create policies that define the types of data you want to mask and the rules that you might use to mask the data.
2. Create a project and import data sources.
3. Optionally, discover information about the source data. Run profiles for data and metadata discovery to discover primary keys, entities, and data domains.
4. Define data subset operations and data masking operations. Define the tables that you want to include in the subset database and the relationships between the tables. Assign data masking rules to columns in the source data.
5. Define data generation operations. Define the tables that you want to include and assign data generation rules to columns in the target table.
6. In addition to the target connection, to directly copy flat file results to an HP ALM server, enter the test tool integration properties in the plan.
7. To store data in the test data warehouse, select the test data warehouse as the target in the plan.
8. Generate and run the workflow for data masking, data subset, or data generation.
9. Monitor the workflow.

Create a Data Masking Policy

Design policies to mask specific types of data. A policy includes the data domains that describe the data that you want to mask. A policy does not contain any data source. You can apply a policy to more than one project in Test Data Manager.

Define data domains to group sensitive fields by column name or by the column data. Define patterns in the column name or the column data using regular expressions. A data domain also contains masking rules that describe how to mask the data.

To design a data masking rule, select a built-in data masking technique in Test Data Manager. A rule is a data masking technique with specific parameters. You can create data masking rules with mapplets imported into TDM from PowerCenter.

Create a Project and Import Metadata

Create a project to organize the components for data discovery, masking, subset and generation operations.

Import data sources in the project. Create a target schema. TDM overwrites any data that already exists in the target schema. Import metadata for the sources on which you want to perform data subset or masking operations. Import target metadata to perform data generation operations. You can import metadata from a PowerCenter folder or an external database source.

When you import PowerCenter source metadata, the TDM Server sends a request to the PowerCenter Repository Service to extract source metadata from the PowerCenter repository. The PowerCenter Repository Service loads the source metadata to the TDM repository. When you import external database metadata, the TDM Server extracts metadata from the source tables and loads it into the TDM repository.

Discover Source Information

You can run profiles to discover primary and foreign key data, entity relationships, and data domains in source tables.

When a data source has no keys, you can run a primary key profile to identify possible primary keys. When the project contains multiple sources, you can run an entity profile to discover possible relationships between tables. Select the primary keys and the entities from the profile results to define the subset data structure.

You can run a data domain profile to search for columns in the source data to add to each data domain. Use data domain profile results to determine which columns to mask with the same masking rules.

When you run profiles for data discovery, the TDM Server sends a request to the Data Integration Service to extract data from the source tables. The Data Integration Service loads the profile results to the profiling warehouse. When you add constraints to tables, the TDM Server stores the constraints in the TDM repository. The TDM server does not update the data sources.

Define Data Masking and Data Subset Operations

To define data subset operations, define the tables that you want to include in the subset database and the relationships between the tables. To perform data masking operations, create a plan to run the masking operations. Add policies for masking the data. You can also add rules that are not in policies.

Perform the following tasks in Test Data Manager to define the data masking and data subset operations:

1. Create entities, groups, and templates to define the tables that you want to copy to the subset database. An entity defines a set of tables that are related based on physical or logical constraints. A group defines a set of unrelated tables. A template is an optional component that contains the entities and groups.
2. Assign data masking rules to columns in the data source.
3. Create a data subset plan, and add the entities, groups, and templates to it. For each column in a parent table, you can define criteria to filter the data.
4. Create a data masking plan and assign the policies and rules to the plan that you want to apply.

The TDM Server stores projects, entities, groups, templates, and plans in the TDM repository. When you generate and run workflows from plans, the PowerCenter Integration Service runs the workflows and loads the data into the target database.

Define a Data Generation Operation

To perform a data generation operation, create a data generation plan. Add tables and entities to the plan.

Perform the following tasks in Test Data Manager to define the data generation operation:

1. Create entities that you want to add to the generation plan.
2. Create data generation rules and assign the rules to the columns in the target table.
3. Create a data generation plan, and add the entities and tables to the plan. Assign default data generation rules to the columns that do not have rule assignments.

Create a Plan for Data Masking and Data Subset

Create a plan for the data masking and data subset operations. A plan includes the components that you need to generate a workflow. You can combine a data masking and a data subset operation in the same plan,

or you can create separate plans. To save the results to an integrated HP ALM server or to store the results in the test data warehouse, select the appropriate properties in the plan.

1. Create a data subset plan and add the entities, groups, and templates to it. You can define additional criteria to filter the data.
2. Create a data masking plan and assign the policies and rules to the plan that you want to apply.
3. To store the results in the test data warehouse, select the test data warehouse from the list of target connections in the plan.
4. To also copy flat file target results to an integrated HP ALM server, enter the test tool integration properties in the plan.
5. Generate a workflow from the plan.
6. Run the workflow.

When you generate and run workflows from plans, the PowerCenter Integration Service runs the workflows and loads the data into the target database.

Create a Plan for Data Generation

Create a plan to perform data generation operations. You cannot combine a data generation operation with a data masking or a data subset operation in the same plan. You must create a separate plan for data generation. To save the results to an integrated HP ALM server or to store the results in the test data warehouse, select the appropriate properties in the plan.

Perform the following tasks when you want to create a data generation plan:

1. Create a data generation plan and add the tables and entities. You can define additional criteria to filter the data.
2. Enter the number of records that you want to generate.
3. To store the results in the test data warehouse, select the test data warehouse from the list of target connections in the plan.
4. To also copy flat file target results to an integrated HP ALM server, enter the test tool integration properties in the plan.
5. Generate a workflow from the plan.
6. Run the workflow.

Monitor the Workflow

Monitor workflow progress and monitor progress and logs of other jobs such as importing metadata and profiling in the Monitor view. Each workflow appears as a job in the Monitor view.

Access the Monitor view to determine the status of the workflow jobs. You can run the Row Count Report on a successfully run workflow to view the number of rows that a plan affects. View the workflow job status. Access the TDM job log to troubleshoot problems.

TDM Example

An organization wants to enforce a policy to mask sensitive employee stock data in a large data processing environment.

The IT department needs test data for a new employee stock plan in an organization. The organization must ensure that the sensitive data is not compromised in the test data. The test database must contain representative data from the various application environments, including employee personal data, salary data, stock purchases, and job information. Multiple test teams must be able to access the test data and replace modified test data with the original test data when required. The organization uses TDM to establish and enforce a policy for creating the data in the test environment and to store and reuse the test data in the test data warehouse.

The organization completes the following steps:

1. Create a policy. The compliance officer determines the type of employee data that should be masked. The compliance officer creates an Employee_Stock policy.
2. Define data domains. The compliance officer defines data domains to group similar fields for data masking. For example, the data contains columns called Employee_Salary, Yearly_Salary, and Salary_History. All columns that contain "Salary" in the name belong to the same data domain. All columns in the same data domain can receive the same data masking rules.
3. Define data masking rules. The compliance officer creates data masking rules to mask the employee data. For example, the compliance officer masks employee names with substitution masking from a dictionary. The compliance officer applies random masking to the salary columns. He applies Social Security masking to Social Security numbers.
4. Define a project. A project developer defines an Employee_Stock project and imports the data sources to the project. The project developer performs all the data subset, data profiling, and data masking configuration in the project.
5. Run a profile for data discovery. The project developer runs a profile for data discovery. The profile identifies sensitive columns in the source tables and it populates the data domains that the compliance officer defined in the policy.
6. Create table relationships. The database does not contain primary and foreign keys. The project developer runs a profile for primary keys and entities to find relationships between tables. The project developer examines the primary key profile results and the entity profile results to create relationships. The project developer creates logical primary and foreign keys in the tables. In some cases, the project developer selects an entity to use from the profile results.
7. Create entities and groups for data subset. With the constraints in place, the project developer can create entities in an Employee_Stock project. An entity defines a set of related source tables based on constraints. The project includes the Employee, JobHistory, Salary, and Employee_Stock tables. The project developer also creates a group in the project. A group defines unrelated tables to include in the test database. The group includes a table called Stock_History.
8. Approve or reject profile job results. The compliance officer reviews the results and approves or rejects the column assignments to the data domains.
9. Verify all sensitive fields are masked. The compliance officer reviews reports that describe what source data is masked in the project.
10. Create a plan to run data subset and data masking. The project developer creates one plan to run the data masking and subset operations in a workflow. The project developer adds the entities and groups to the plan to define which data to copy to the subset database. The project developer adds the Employee_Stock policy to the plan to define how to mask the data. When the project developer runs a workflow from the plan, the PowerCenter Integration Service runs the workflow and loads the masked data into the subset database.

11. The compliance officer validates the results in the subset database.
12. Create a plan to move the masked data subset to the test data warehouse. The project developer creates a plan with the subset database as the source connection and the test data warehouse as the target connection. When the project developer runs a workflow from the plan, the PowerCenter Integration Service runs the workflow and loads the masked data as a data set in the test data warehouse.
13. Reset a data set from the test data warehouse. The project developer runs a reset operation on the data set to restore the original test data to the required connection. When the reset operation runs, the PowerCenter Integration Service runs the workflow and loads the data set from the test data warehouse to the target connection.

Using Advanced Parameterization in Test Data Manager

You can use parameters to create multiple test data results from the same plan.

Consider a test team that uses TDM to create test data for different projects. The team follows the same process to create test data from production data on each project. The source connection that you use to create the test data differs and the criteria or filter value in the criteria might differ. Each time you might create the test data in a different target schema or target connection. For the same project, you might need to create test data multiple times for specific test cases. You use different filter criteria to meet specific data requirements.

Create connection, owner, and criteria parameters for connections, schema, and to use in filter criteria in an entity. You can then use the same plan to create multiple test data results. Instead of creating different plans or editing the plan each time, you can use the same plan and change the parameter values for connections, owners, and criteria. You can also choose to include or exclude specific criteria parameters each time you run a workflow to create different results with the same data.

If you update one or more parameters or parameter values each time, you can run multiple workflows from the same plan to create different results.

Note: You must import the source metadata from all source connections that you need into the project. You cannot run a workflow that includes a parameter value that points to source metadata that the project does not contain.

To use parameters to create different test data output from the same plan, perform the following high-level tasks:

1. Create a project and import the required source metadata.
2. In the project, create parameters for connections, owners, and criteria. Assign default values and choose to allow override of the default value.
3. Create an entity that includes the required tables.
4. Add filter criteria that uses parameters. You can add multiple criteria and criteria parameters.
5. Create a plan that uses parameters.
6. Optional. In the **Criteria** page of the plan, edit the entity criteria parameters as needed.
7. In the **Plan Settings** page, add the source and target connection types as variables, and select the required connection parameters for each. Enter other properties as needed. You must enter a parameter file name and the path to the file on the system that hosts the PowerCenter Integration Service.

8. In the **Advanced Settings** section, add the schema owners as variables and select the required parameters.
9. Save the plan.
10. Optional. From the plan **Properties** tab, click **Actions** > **Create Parameter File** to download a file that lists all parameters included in the plan.
11. Optional. Review the list of parameters in the file. You can edit the values and save the file. If the entity includes multiple criteria parameters, you can choose to exclude specific parameters and include others. Update the values of the ignore flags to "Y" to exclude criteria parameters. You can choose to use the file when you run the workflow.
12. From the plan **Properties** page, click **Actions** > **Generate and Execute**.
13. Choose the parameters and parameter values to include in the workflow.
    - To use parameter values in a parameter file that you maintain on the machine that hosts the PowerCenter Integration Service, choose to use the default parameter file. You enter this file path and file name in the **Plan Settings** page of the plan creation wizard. Choose **Default** to use the file entered in the **Plan Settings** page.
    - To use parameter values from a different parameter file, choose **Use Parameter File** and browse and select the required file. For example, you might create a parameter file in Test Data Manager and update parameter values in the file. You can choose this option to use parameters from this file.
    - To update or use parameters from Test Data Manager, choose **Add Parameter Values**. A list of parameters in the plan appears with default values. You can edit the values. Click the **Select Parameter** button to select and include criteria parameters in the workflow. You can update the values of criteria parameters that you add to the list.
14. Click **Generate and Execute**.
15. Monitor the workflow.

You can update parameters as required to create different test data from the same plan.

For example, you can edit parameter values and run the same plan with the following changes:

- Create test data from the same source connection and in the same target connection but different schema and with different criteria values. Update the owner parameter values and criteria parameter values and run the workflow.
- Create test data from and to different source and target connections with different criteria. Update the connection parameter values and choose to include different criteria parameters in the workflow.

Each time you run a workflow with different parameters or parameter values, the test data that you create differs. You can create different test data without creating multiple plans or editing the plan.

Related Topics:

- "Parameters" on page 50
- "Creating a Parameter" on page 54
- "Editing or Deleting a Parameter" on page 55
- "Executing a Workflow" on page 295

CHAPTER 2

Test Data Manager

This chapter includes the following topics:

- Test Data Manager Overview, 36
- Test Data Manager User Interface, 37
- Overview View, 40
- Policies View, 42
- Data Sets View, 42
- Projects View, 43
- Monitor View, 44
- Administrator View, 44
- Expression Builder, 45
- Logging In to Test Data Manager, 46

Test Data Manager Overview

Test Data Manager is a web-based user interface that you use to perform data discovery, data subset, data masking, and data generation operations. Manage data sets in the test data warehouse from Test Data Manager. Open a view in Test Data Manager based on the task you need to perform.

A compliance officer uses Test Data Manager **Policies** view to create policies, data masking rules, data generation rules, and data domains. The compliance officer assigns the data masking rules to data domains.

A developer uses the **Projects** view in Test Data Manager to define a project and import the data sources to the project. The developer runs profiles to discover primary keys, entities, and data domain assignments. The developer creates entities for data subset operations and assigns rules to columns for data masking and data generation operations. The developer creates plans, generates workflows, and runs the workflows.

The developer opens the **Monitor** view to check the status of jobs that perform data masking, data subset, data generation, and other operations.

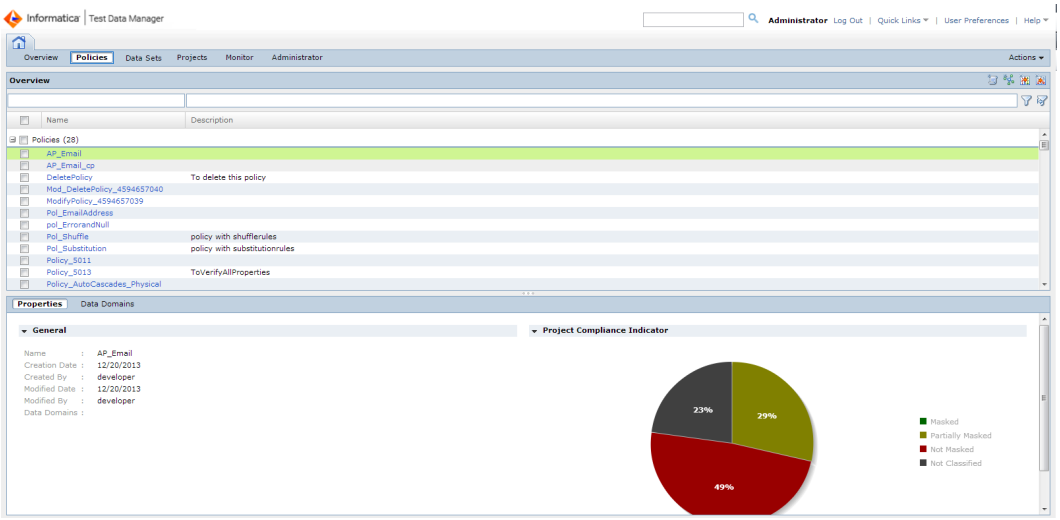
A test data specialist or a test engineer uses the **Data Sets** view to view and manage data sets in the test data warehouse.

An administrator uses the **Administrator** view to create connections, create and manage pass phrases, configure workflow options, perform server management, and restrict user access to TDM components. An administrator can also integrate a test tool with TDM and configure the test data warehouse from the **Administrator** view.

Test Data Manager User Interface

Test Data Manager contains options to view and edit TDM components.

The following image shows a view in Test Data Manager:



The Contents panel shows an overview of the items in a view. The Details panel shows additional details for a single item in the contents panel.

Views

Access Test Data Manager views to perform tasks such as defining data masking policies or configuring projects.

Test Data Manager contains the following views:

Overview

View dashboard reports about projects in the TDM repository.

Policies

Define policies, masking rules, and generation rules that you can add to projects.

Projects

Define a project that contains source data and the data subset, data masking, data profiling, or data generation operations for the data.

Data Sets

Manage data sets that you store in the test data warehouse.

Monitor

View the status of jobs that import sources or perform data subset, data masking, data profiling, or data generation operations. Stop or abort jobs.

Administrator

Manage connections, pass phrases, dictionaries, test tool integration, test data warehouse, and workflow options.

Note: By default, an administrator can access the **Administrator** view of Test Data Manager. A user must have privileges to access the other views in Test Data Manager.

Search Field

Use the Search field to search for objects within the TDM repository and the test data warehouse repository. Search for objects such as projects, plans, assignments, data sets, and data set tables. You cannot search for connections.

Enter the name or part of the name of the object you want to find. A search tab opens where you can filter the types of objects to include in the search. A search count displays the number of objects that match the search criteria. The search results return all objects and assignments contained within it and objects that contain it. For example, when you include projects within the search objects, the results also list objects such as policies, rules, and entities that the project contains. When you search for a masking or generation rule, the results include the rule, assignments, objects that contain the rule, and dictionaries in the rule.

You can search for objects other than connections globally from the Search field. You cannot search for objects when indexing tables.

Advanced Search Text Filter

In Test Data Manager, you can enter a text instead of the wild characters in the filter criteria to search for columns.

Based on the type of filter operators, TDM searches and displays the results.

The following table lists the sample formats that the filter supports:

Filter Type	Sample Text	Operator Type
Exact match	"EMPLOYEE_FIRSTNAME"	Equality operator
Starts with	EMPLOYEE*	Like operator and type is STARTS_WITH
Ends with	*SALARY	Like operator and type is ENDS_WITH
Contains	BONUS	Like operator and type is CONTAINS

The following table lists the Test Data Manager views and pages in which you can use the advanced filter:

Test Data Manager Page	Path in Test Data Manager	Advanced Filter Supported Columns
Discover | Column	Project -> Discover -> Columns	Table Name, Column, Owner, Data Type, and Domain
Define | Data Masking	Project -> Define -> Data Masking	Name, Owner, Columns, Data Type, Domain, Policy, and Masking Rule
Define | Data Generation	Project -> Define -> Data Generation	Table, Owner, Column, Data Type, Data Domain
Profile Results	Project -> Discover -> Profile -> Column Profile Results -> Column Table	Table, Column, and Data Type

Quick Links Menu

You can quickly navigate to content tabs in Test Data Manager from the **Quick Links** menu.

From the **Quick Links** menu, you can open tabs to manage data subset, masking, and generation. You can use the **Quick Links** menu to access shortcuts to create a generation rule, masking rule, policy, data domain, or project.

The items that you can access from the **Quick Links** menu change based on user permissions.

User Preferences Menu

The **User Preferences** menu contains options that you can use to modify Test Data Manager.

Click **User Preferences** to view the **Edit Preferences** dialog box. The **Edit Preferences** dialog box contains the following tabs:

General

Displays general Test Data Manager preferences. Select **Show Start Screen** to display the **Informatica Test Data Manager** start screen when you log in to Test Data Manager.

Projects

Displays a list of projects that are visible in the dashboards in the **Overview** view. Disable projects in the **Projects** tab that you do not want to appear in the **Overview** view.

Actions Menu

Click the **Actions** menu to modify data that you select in the Content panel or to perform tasks such as importing and exporting objects.

You can choose different options in the Actions menu based on the Test Data Manager view.

Data Filter

You can filter data that appears in the **Content** and **Details** panel of Test Data Manager views.

Test Data Manager **Content** and **Details** panels might contain multiple rows of data. For example, the Policies view might show 25 data domain names. You might want to limit the data domain names to names that include "Emp." To apply the filter, type "Emp" in the filter field for the data domain name. Click the **Filter** icon. The **Policies** view shows data domains such as Employee_Domain or Emp_Number.

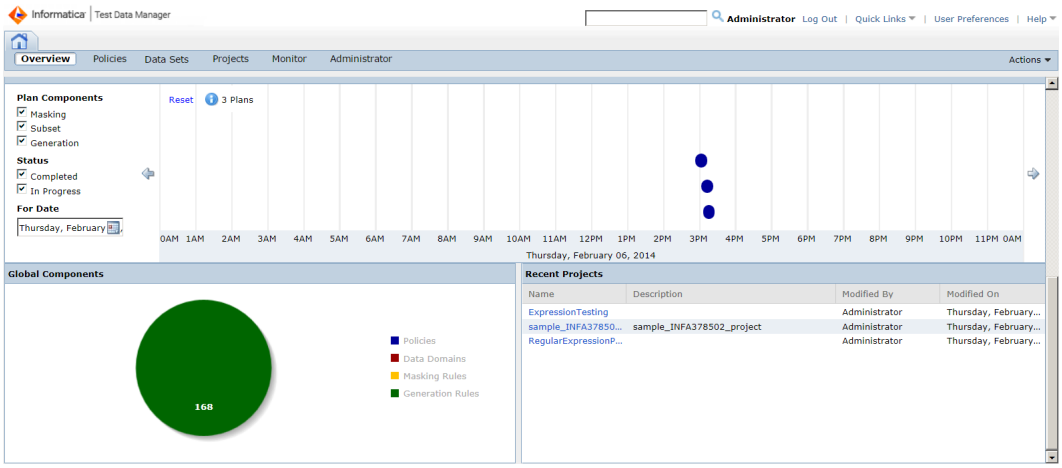
To reset the filter results, click the **Reset Filter** icon.

You can apply filters to different columns of data in the **Contents** panel based on Test Data Manager view.

Overview View

View dashboard reports about projects in the **Overview** view. The dashboard reports are graphs that show the distribution of data masking, data subset, and data generation objects in projects. You can use links in the **Overview** view to access project and the policy objects.

The following image shows the **Overview** view:



Dashboards

The **Overview** view shows the TDM dashboards. The dashboards are summary reports about projects, data masking objects, data generation objects, and administration objects.

You can view the distribution of data by sensitivity level in projects. You can view the distribution of data masking, data domains, policies, and rules in projects. View the distribution of entities, groups, and templates in data subset projects.

You can view the number of data masking policies, the number of data domains, and the number of rules in the TDM repository.

Project Risk Analysis

The **Project Risk Analysis** dashboard shows the percentage of columns that are in each domain sensitivity level. It also shows the percentage of columns that do not belong to domains.

When you create a data domain, you select the sensitivity level for all columns in the domain. You can select a sensitivity level from the levels configured by the administrator.

The **Project Risk Analysis** dashboard shows the percentage of columns that belong to domains at each sensitivity level. You can view the projects that contain the columns. Move the pointer over the section of the pie chart that you want to view. The dashboard shows the number of domains in that sensitivity level, and the number of projects that contains columns in the domains. Click the **Number of Projects** link to view a list of the projects.

Project Assignments and Project Objects

The Project Assignments and Project Objects dashboard shows graphs of the number of objects and assignments in up to 10 recently accessed projects.

The Project Assignment tab displays by default. This graph displays the number of assignments, including data domains, policies, masking rules, and generation rules, in the last 10 accessed projects. The Project Objects tab displays the number of subset objects, including entities, groups, and templates, in up to 10 recently accessed projects.

Recent Project Sensitive Data Distribution

The **Recent Project Sensitive Data Distribution** dashboard shows a graph of the sensitive domain information in the last 10 accessed projects.

You must have set data domain sensitivity levels to view the distribution information. If you do not set data domain sensitivity levels, no graph appears.

Recent Activities

The **Recent Activities** panel shows the last 10 Test Data Manager components that changed.

The **Recent Activities** panel shows the name of the TDM component that changed, the type of component, and who made the change. The change actions are created, updated, and deleted. The panel shows the date of the change.

You can view and download an Audit Trail report from the **Recent Activities** tab.

Plan Execution

The **Plan Execution** dashboard displays plan execution data for a particular date.

You can select the plan component, masking, subset, or generation, for which to view data. You can view data for plans in progress, completed, or both. The default display is for the current date, all plan components, and both completed and in progress status. You can browse the calendar to choose a different date, or enter the specific date. Use the arrows to move the date forward or backward.

Global Components

The **Global Components** dashboard shows the proportion of policies, masking rules, generation rules, and data domains, out of the total number of components available in the repository.

Use the **Global Components** dashboard to view the distribution of policies, data domains, masking, and generation rule assignments in the repository in a pie chart. Masking rules and generation rules are displayed separately.

Recent Projects

The **Recent Projects** dashboard lists up to the last 10 projects you modified. You cannot view information on projects created or modified by other users.

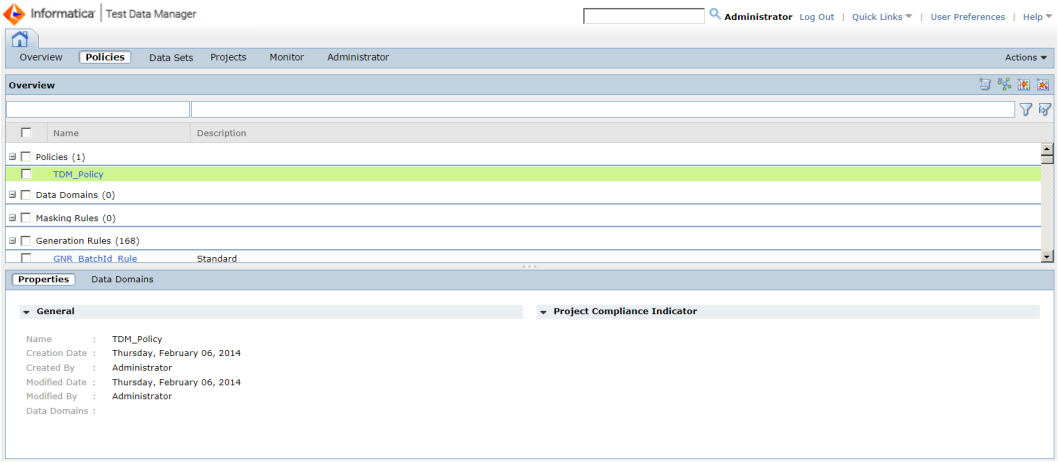
The **Recent Projects** dashboard shows the name of the modified project, the project description, the user name, and the date of modification. Click on the project name to open the project.

Policies View

Maintain policies, the data domains, and the data masking and generation rules in the **Policies** view. Create a policy and add the data domains to the policy. You can apply the policy to multiple data masking projects in Test Data Manager.

The **Policies** view lists the policies, the data domains, and the rules in the TDM repository. Select a policy, a data domain, or a rule to change it. Or, choose to add a new policy, data domain, or rule, from the **Actions** menu.

The following image shows the **Policies** view:



Data Sets View

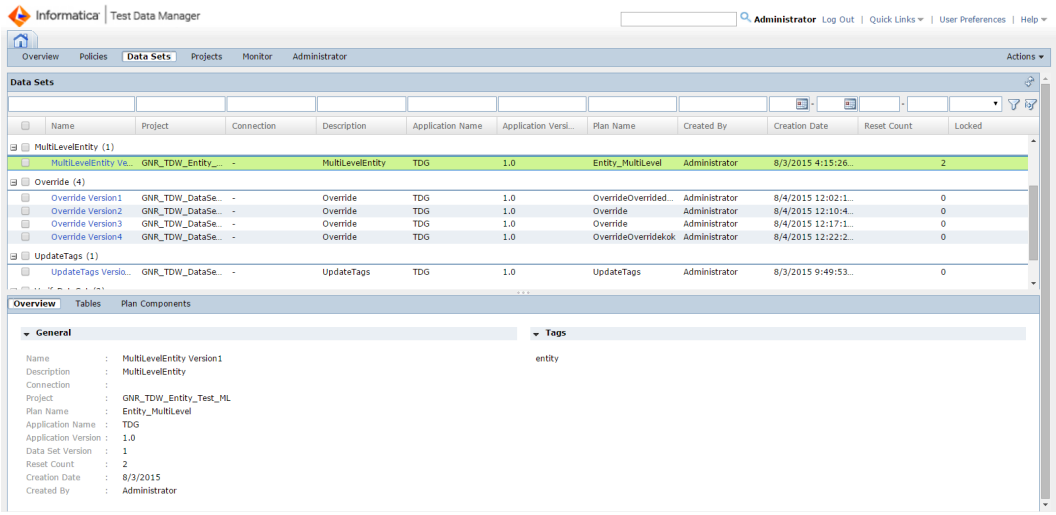
Manage multiple versions of data sets from the **Data Sets** view.

The **Data Sets** view lists all data sets stored in the test data warehouse. Select a version of a data set from the list to view the general properties in the **Overview** tab. The **Tables** tab lists all tables in the data set that you select. The **Components** tab lists the components in the data set.

Use the **Data Sets** list filter to filter and search for a data set. You can reset a specific version of a data set to a target, delete a data set, or lock and unlock a data set from the **Actions** menu.

You can open a data set in a separate page and view and edit the data in the data set.

The following image shows the **Data Sets** view:



Projects View

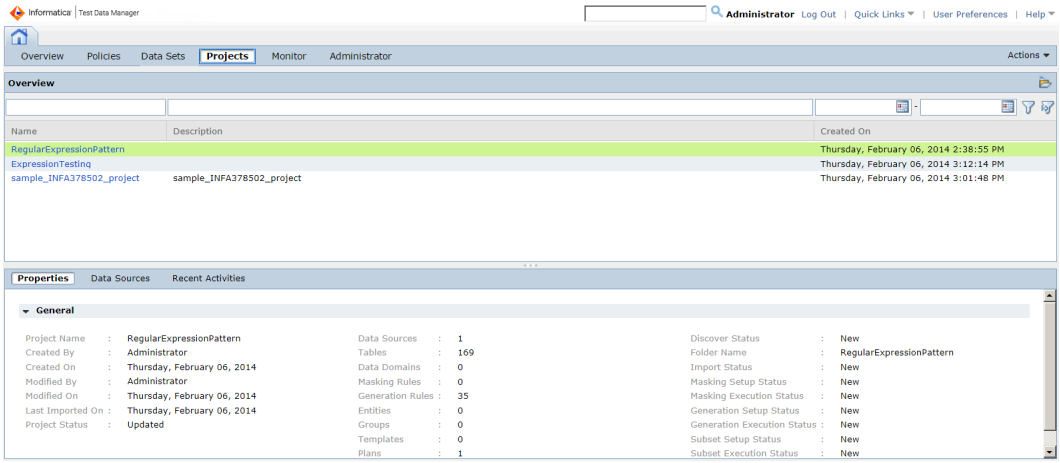
Maintain projects in the **Projects** view. A project is the top-level container that you use to organize the components for data discovery, data masking, data subset, and data generation operations.

From the **Projects** view, you can create a project and you can import source data for the project. Assign policies and rules to columns in the source data. Run profiles for data discovery against the data. Run data masking and data subset operations on the source data. Run data generation operations to generate data.

The **Projects** view shows a list of projects. You can select a project to view the data sources and recent activities.

When you select a project, you can update it. You can define profiles for data discovery, data masking, and data subset operations. You can generate a PowerCenter workflow from within a project.

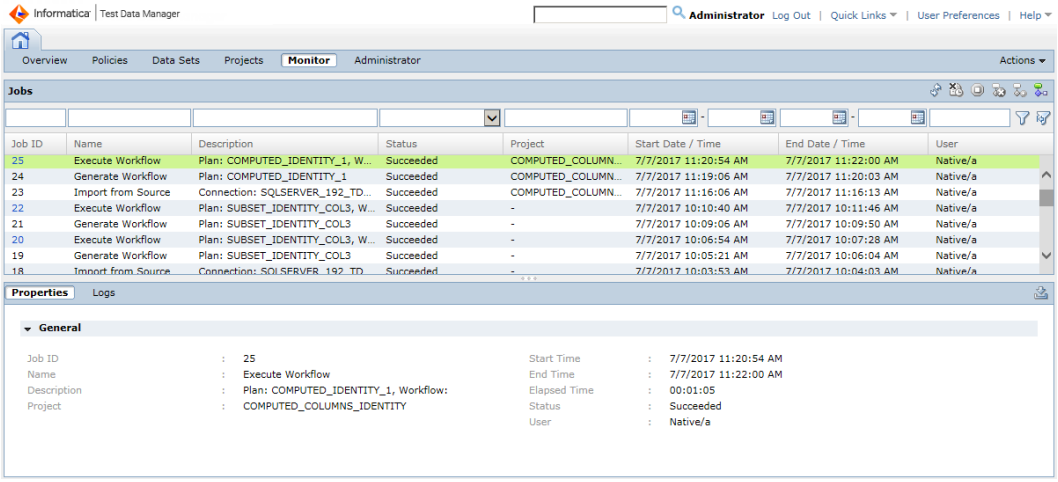
The following image shows the **Projects** view:



Monitor View

In the **Monitor** view, you can review the status of jobs that perform tasks such as import data, run profiling, or run data masking, data subset, and data generation operations. You can review the TDM Server job log to investigate problems. You can also stop jobs and recover workflows in the **Monitor** view.

The following image shows the **Monitor** view:



Select a job in the **Monitor** view and choose options from the **Actions** menu.

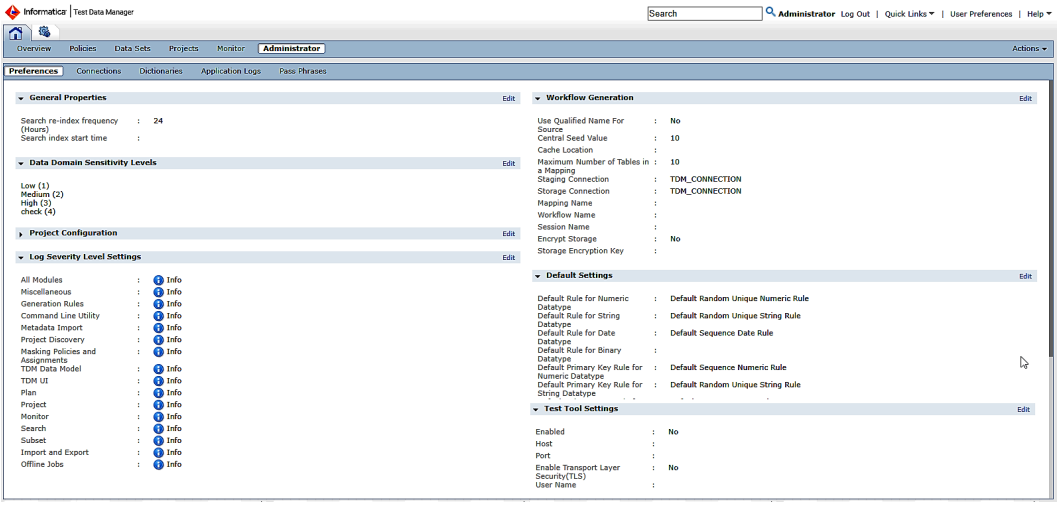
Administrator View

An administrator can set default settings, create connections, add and manage dictionaries, and perform TDM server tasks in the **Administrator** view.

An administrator can perform the following tasks in the **Administrator** view:

- Configure default settings including generation rule settings and child and parent record settings.
- Configure data domain sensitivity levels for tracking the sensitive data that users need to mask.
- Add and modify default project configuration fields.
- Configure workflow properties and set default staging and storage connections and create storage tables.
- Create connections to source or target databases.
- Add and manage dictionaries.
- View and search application level logs.
- Set or modify the log levels to debug the logs.
- Integrate a test tool with TDM.
- Create and manage pass phrases.

The following image shows the **Administrator** view:



Expression Builder

Use the **Expression Builder** to build an expression using PowerCenter functions.

Click **Expression** to open the **Expression Builder** when you perform any of the following tasks:

- Create masking rules with Expression masking type.
- Add pre-processing or post-processing expressions to a standard masking rule.
- Create advanced masking rules.
- Assign rules to a column.
- Configure expressions in an ad hoc generation rule.
- Configure advanced expressions in plans.

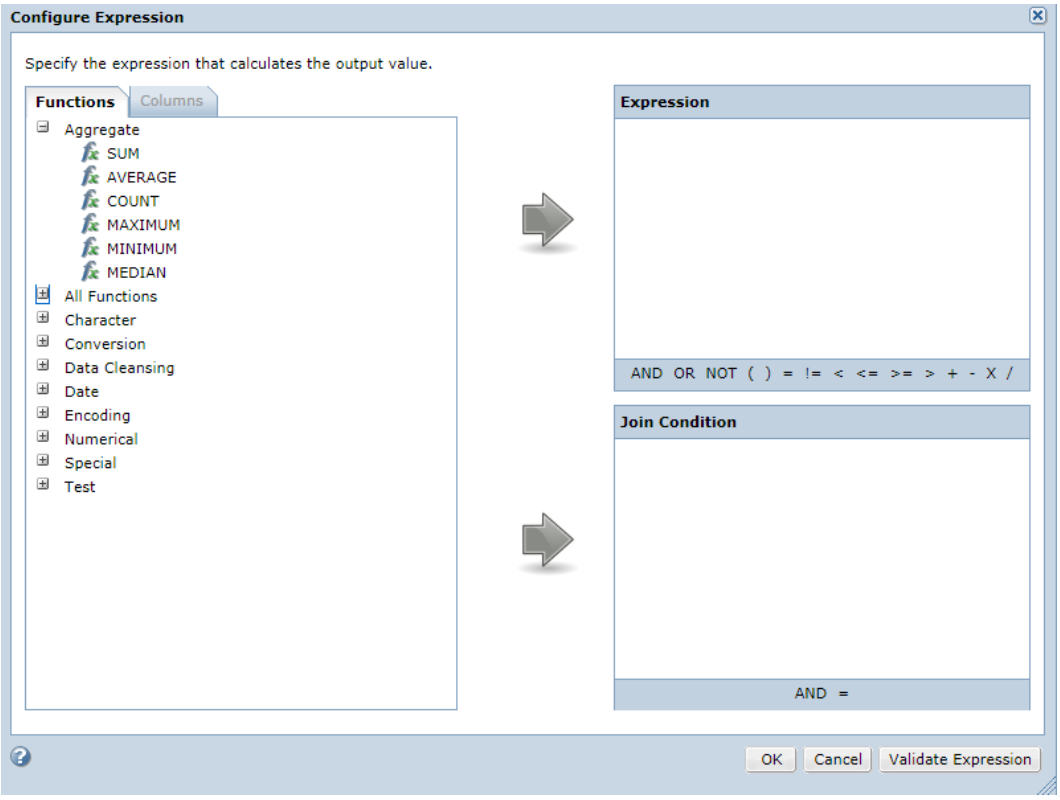
The **Expression Builder** shows a list of PowerCenter functions by category. To view the complete list, expand the **All Functions** list.

You can choose from a list of available columns to include in an expression. Select a function and click the **Add** arrow to add the function to an expression. You can choose columns and operators to enter join conditions.

For example, you might create an advanced masking rule that contains a first name and a last name input port. The masking rule has two variable ports that receive masked values for the first name and the last name. You might create an expression for the output port that concatenates the masked names in the variable ports and includes a space between them:

```
CONCAT( CONCAT( VAR1, ' ' ), VAR2 )
```

The following image shows the **Expression Builder** where you configure expressions:



For more information about the function syntax, see the *Informatica Transformation Language Reference*.

Logging In to Test Data Manager

To access Test Data Manager, enter the host name and port number of the TDM Server in a web browser.

To log in, enter a user name and password defined in Informatica Administrator.

1. In the address bar of a web browser, enter the Test Data Manager URL.
   - Use the following format if Transport Layer Security is enabled:

     `https://hostname:portnumber/tdm/`
   - Use the following format if Transport Layer Security is not enabled:

     `http://hostname:portnumber/tdm/`

   Where:
   - *hostname* is the host name or IP address of the machine where you installed the TDM Server.
   - *portnumber* is the port number. The default is 6643 if Transport Layer Security is enabled. The default is 6605 if Transport Layer Security is not enabled.

   For example, you might enter the following URL:

   `http://TXW1779:6643/tdm/`

   The **Login** dialog box of Test Data Manager appears.
2. Enter the user name and password.

Select the security domain. If the Informatica domain is configured to use LDAP authentication, the default security domain **Native**.

3. Click **Login**.

   Test Data Manager opens.

To log out of Test Data Manager, click **Logout**.

CHAPTER 3

Projects

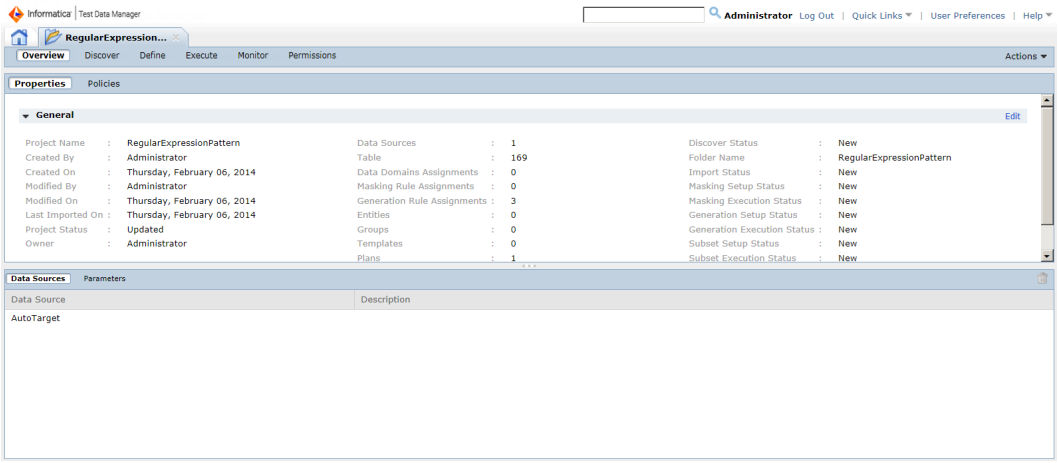
This chapter includes the following topics:

- Projects Overview, 48
- Project Components, 49
- Parameters, 50
- Project Logs, 51
- Data Masking Transformation Session Properties, 51
- Project Management, 52
- Data Sources, 56
- Delete a Table, 62
- Project Permission and Security, 62

Projects Overview

A project is the top-level container that you use to organize the components for data discovery, masking, subset, and generation operations.

The following image shows a project in Test Data Manager:



A project contains the following views:

Overview

Edit the project general properties in the **Properties** view. Add policies and rules to the project in the **Policies** view. In the **Data Sources** view, import data sources to a project for data discovery, data subset, data masking, and data generation operations. You can import a source from the PowerCenter repository, or you can import a source from a database. You can import multiple types of sources to the same project and define relationships between them. In the **Parameters** view, create and maintain parameters for the project. You can create connection, owner, and criteria parameters.

Discover

Discover the primary keys, the table relationships, and the data domains in the source data.

Define

Define data masking, data subset, and data generation operations. When you define a data masking operation, assign the rules and policies to columns in the project source. When you need to create a data subset, define entities, groups, and templates. When you need to perform a data generation, assign the generation rules to the columns in the tables of the project.

Execute

Define a data subset, data masking, or data generation plan. Generate and run a PowerCenter workflow from the plan.

Monitor

View the status of data source import jobs, profiling jobs, workflow generation jobs, data subset jobs, data masking jobs, and data generation jobs. You can refresh and abort a job. You can abort, stop, and recover a workflow. You can remove a job that is scheduled to run at a later time.

Permissions

Apply the user group and the user permission to projects. When you create a project, you are the project owner and have access to the project. If other users need access to the project, you can add the users in the **Permissions** view. Apart from permission on a specific project, users need the minimum required privileges to access or perform any tasks on the project.

Project Components

A project contains one or more data sources. Other components that you add to a project depend on the operations that you need to perform on the data.

The following table describes the components that you can create in a project:

Component	Operation	Description
Data Source	Data subset Data discovery Data masking Data generation	Defines the input data that you want to transform.
Entity	Data subset Data generation	Defines a set of related tables based on physical or logical constraints. Define an entity for a data subset or a data generation operation. Create an entity in the **Define** view of a project.

Component	Operation	Description
Group	Data subset	Defines a set of unrelated tables to copy to a target subset database with data subset. Create a Group in the **Define** view of a project.
Parameters	Data subset Data masking Data generation	Defines parameters and parameter values for connections, criteria, and source or target owners.
Plan	Data masking Data subset Data generation	Defines data subset, data masking, or data generation operations. Generate a PowerCenter workflow from a plan. Define a plan in the **Execute** view of a project.
Policy assignments	Data masking	Assigns policies to the project. When you assign a policy to a project, the project receives the data domains and the data masking rules in the policy.
Profile	Data discovery	Suggests the primary keys, relationships, and data domains in a source based on source data and metadata. Define a profile in the **Discover** view of a project.
Rule assignments	Data masking Data generation	Defines which data masking or generation rules apply to a source column.

Parameters

You can create parameters in Test Data Manager. Use parameters to vary the output of a workflow without updating the plan.

Parameters work as placeholders that point to a value. When you add a parameter in a plan, the workflow uses the value that the parameter points to at run time. You can change the value to change the value that the workflow uses.

You can create parameters for connections, owners, and entity criteria. Add a default value when you create the parameter. You can override the value and enter different values or use different criteria parameters in the workflow to create different data each time you run the workflow.

You can create the following types of parameters:

- Connection parameter. A placeholder for source or target connections.
- Owner parameter. A placeholder for source or target schema owner names.
- Criteria parameter. A placeholder for values that you include in filter criteria in an entity.

Use criteria parameters when you create an entity and select the required parameter in the workflow.

Select parameters from the list of available parameters when you create a plan. Before you run a workflow, you can create a sample parameter file that contains all parameters included in the plan. Choose to use the parameter file as is or to edit the parameter values and use the file in the workflow. You can also edit parameter values in Test Data Manager when you run the workflow.

Rules and Guidelines for Parameters

You can create parameters in a project and then use the parameters in TDM operations.

Consider the following rules and guidelines when you use parameters to perform TDM operations:

- If you use a connection parameter for an Oracle source connection, you cannot enable the Oracle *Rowid* property.
- You cannot perform batch updates for inplace masking if you use connection parameters because a batch update requires a staging connection. You cannot use a parameter for a staging connection.
- If you use connection parameters in a plan and select the same connection parameter as source and target, the **Update Strategy** property value does not change to *Update*.
  You can choose to use one of the following workarounds:
  - Select the **Use Source Connection as Target Connection** option. When you select the option, the **Update Strategy** property value changes to *Update*.
  - Manually change the value of the **Update Strategy** property to *Update*.

Project Logs

PowerCenter generates workflow and session logs when you run a plan or workflow from within a project.

You can configure the location where you want to save the log files for a project. If you do not specify a location, PowerCenter saves the log files in the default location set in the PowerCenter Integration Service properties.

The PowerCenter Integration Service stores the logs and reject files in the following default locations:

- Workflow log directory: `$PMWorkflowLogDir\`
- Session log directory: `$PMSessionLogDir\`
- Reject file directory: `$PMBadFileDir\`

You can configure the location where you want to save workflow logs, session logs, and reject files from the **Edit Project** dialog box. You can specify the location in an existing project. You cannot specify the location when you create the project.

You can configure workflow log properties and session log properties when you create a plan. For example, you can choose to save session logs or workflow logs by the time stamp, the number of runs, or the maximum size of the log files.

You can specify the type of error log files to generate error logs and the location to store error logs for a specific plan when you create the plan.

Data Masking Transformation Session Properties

You can configure the session run-time properties to increase performance.

Configure the following session run-time properties:

SSN High Group File

The path to the High Group List directory that contains the Social Security numbers issued by the Social Security Administration.

DMO Cache Directory

The location of the dictionary cache. You must have write permissions for the directory. Default is $PMCacheDir.

Cache Size

The size of the dictionary cache in main memory. Increase the memory size in order to improve performance. Minimum recommended size is 32 MB for 100,000 records. Default is 8 MB.

Shared Storage Table

Enables sharing of the storage table between Data Masking transformation instances. Enable Shared Storage Table when two Data Masking transformation instances use the same dictionary column for the database connection, seed value, and locale. You can also enable the shared storage table when two ports in the same Data Masking transformation use the same dictionary column for the connection, seed, and locale. Disable the shared storage table when Data Masking transformations or ports do not share the dictionary column. Default is disabled.

Storage Commit Interval

The number of rows to commit at a time to the storage table. Increase the value to increase performance. Configure the commit interval when you do not configure the shared storage table. Default is 100,000.

Project Management

When you create a project, you add one or more sources to the project. When you edit a project, you can update the name, description, folder location, and associated sources. You can also edit the location where you want to store the log files of the project.

When you create a project, you are the project owner. You have access to the project folder by default.

You can export a project and project components to an XML file. You can import the data from the XML file into Test Data Manager. You might export and import a project to back it up, or you might need to move a project from a development to a production system.

Creating a Project

Create a project folder to contain the data discovery, subset, masking, and generation components to apply to a data source.

1. In Test Data Manager, click **Projects** to access the projects.

   A list of projects appears.

2. Click **Actions** > **New**.

3. In the **Create Project** dialog box, enter project properties. The following table describes the project properties:

Option	Description
Name	The name of the project.
Description	The description of the project.
PowerCenter Repository	The name of the PowerCenter repository to store the repository folder.
Folder	The name of the project folder in the repository. Default is the project name. You can choose another folder in the repository.
Owner	The name of the user that owns the folder. The folder owner has all permissions on the folder. The default is the name of the user that created the folder. You can select another user as the folder owner.

4. Click **OK**.

   The properties for the project appear in Test Data Manager.

Editing a Project

You can edit a project to change its name, description, or associated sources.

1. To access the projects, click **Projects**.

   A list of projects appears.
2. Click on the project that you want to edit.

   The project opens in a separate tab with project properties and data sources details.
3. To change the project properties, click **Edit**.

   You cannot remove a source that is used by entities or groups in the project.
4. In the **Edit** dialog box, change project options. The following table describes the project options that you can change:

Option	Description
Name	The name of the project.
Description	The description of the project.
Folder	The name of the project folder in the repository. Default is the project name. You can choose another folder in the repository.
Owner	The name of the user that owns the folder. The folder owner has all permissions on the folder. The default is the name of the user that created the folder. You can select another user as the folder owner.

5. Click **Save**.

Copying a Project

You can create a project by copying a project. When you copy a project, Test Data Manager creates a copy of the project but does not import the metadata or objects of the project.

1. To access the **Projects** view, click **Projects**.
2. Click a project description to select the project.

   Do not open the project.
3. Click **Actions** > **Duplicate**.

   The **Copy <Project Name>** dialog box appears.
4. Change the name and description of the project. Click **Save**.

Deleting a Project

When you delete a project, all sources that are associated with the project are deleted.

1. To access the projects, click **Projects**.

   The list of projects appears.
2. To select a project without opening it, click the Description or Created On column of the project.
3. Click **Actions** > **Delete**.
4. In the **Confirm Delete** dialog box, click **Yes**.

Creating a Parameter

Create parameters in a project if you want to use parameters in a plan.

1. Click **Projects** and open the required project.
2. Click **Parameters** to view the **Parameters** tab.
3. Click the **Edit** button to open the **Configure Parameters** window.
4. Click **Add**.
5. Enter a name and optional description for the parameter.

   The naming convention depends on the parameter type.
   - Criteria parameters must begin with '$$'.
   - Connection parameters must begin with '$DBConnection'.
   - Owner parameters must begin with '$Param'.
6. Select the parameter type from the list.

   You can create criteria, connection, and owner parameters.
7. Select the data type.

   You can create a parameter with String, Numeric, or Date data types.
8. If you choose the Date data type, enter an expression to define how a function uses the parameter.

   For example, *TO_DATE($$CRIT1,'MM-DD-YYYY')* where $$CRIT1 is the name of the parameter.
9. Enter a default value. Required if you do not choose to allow override.
10. Choose to allow override of the default value.

    You cannot change the parameter value when you run the workflow if you do not allow override.

11. Click **OK** and then click **Save** to create the parameter.

Editing or Deleting a Parameter

You can update or delete a parameter that you create in a project.

1. Click **Projects** and open the required project.
2. Click **Parameters** to view a list of parameters in the **Parameters** tab.
3. Click the **Edit** button to open the **Configure Parameters** window.
4. You can edit or delete a parameter.
   - To edit a parameter:
     1. Click to select the row without selecting the check box. The fields that you can update become editable.
     2. Edit the values as required. You cannot change the parameter type or the data type.
     3. Click the **Save** button and then click **Save**.
   - To delete a parameter:
     1. Select the required parameter check box from the list.
     2. Click **Delete** and then click **Save**.

   A list of entities and plans that include the parameters appears. To update the changes in the plans, generate and run the plans again.
5. Optional. To download the list in a .csv file, click **Export**.

   The file contains a list of the entities and plans with the description and the project information.
6. Click **Continue**.

Configuring Project Log Locations

You can configure the location where you want to save the log files that the PowerCenter Integration Service generates for project workflows and sessions.

1. Open the project and click **Actions** > **Edit**.

   The **Edit Project** dialog box opens.
2. Click the **Default Log File** tab.
3. Perform the following tasks to change the location of the log files:
   - In the **Workflow Log Directory** field, enter the location where you want to store workflow logs.
     If you enter a wrong location, a workflow fails to run without generating an error log.
   - In the **Session Log Directory** field, enter the location where you want to store session logs.
   - In the **Reject File Directory** field, enter the location where you want to store reject files.

   You can use parameters or enter a fully defined path. You must add a delimiter with a slash or backslash at the end of the file path.
   For example, you can enter the workflow log directory in the following ways:
   - `$PMWorkflowLogDir\`
   - `$PMWorkflowLogDir\projectlogs\`

- `\TDM\Logs\Project\`

Note: The directory path must exist. TDM does not create the directory.

4. Click **Save**.

Exporting a Project

You can export a project to an XML file and import the XML file to another TDM repository. When you export a project, the XML file contains all the project components, such as the source information, the connections, the constraints, the data domains, and the assignments.

1. To access the projects, click **Projects**.

   A list of projects appears.
2. Click a project to view the project properties and the data sources.
3. Click **Actions** > **Export**.
4. Choose the name and path of the XML file to create.
5. Click **OK**.

Importing a Project

You can import a project from an XML file that was exported from another TDM repository.

Before you import a project into Test Data Manager, you must import all the global components that are assigned in the project.

1. To access the projects, click **Projects**.
2. Click **Actions** > **Import**.
3. Browse for the XML file that contains the project to import. Click **Finish**.

   Test Data Manager imports the project.

Data Sources

To perform data subset, masking, and discovery operations, you must import source metadata into the TDM repository. You can import sources from the PowerCenter repository or from a source database. To perform data generation operations, you must import target metadata into the TDM repository.

When you create a project, add one or more sources to the project. You can add more than one type of source to the project. For example, you can add a flat file source and a relational source to the project. You can create constraints to create relationships between the sources and apply filter criteria for data subset and data masking.

Note: If the source data has a numeric column with precision greater than 28, you cannot apply a data masking or data subset condition to that column. The PowerCenter Integration Service passes the row and the column to the target without applying the condition.

TDM Source Definitions

Import source metadata from PowerCenter or directly from the source database to the TDM repository. The TDM project and PowerCenter folders share the same name to ensure that the PowerCenter Integration Service stores workflow information in the correct PowerCenter folder.

You can import source metadata directly from the source if the source is a relational database. If the source contains many tables, you can increase the performance if you import the metadata directly from the source instead of from PowerCenter.

When you import source definitions from the PowerCenter repository into the TDM repository, you can import all the definitions supported by PowerCenter. When you import source metadata from PowerCenter, you import the PowerCenter folder that contains source metadata. Test Data Manager creates a copy of the folder in the TDM repository.

When you import the MongoDB sources to PowerCenter, you must ensure that the column names are unique and does not contain more than 80 characters in a column name.

To use flat file sources, you must import flat files from the PowerCenter repository. You cannot create flat file connections in Test Data Manager.

Note: Rowid is a reserved keyword. You cannot import Oracle sources that have rowid as a column name or a table name.

Hadoop HDFS Data Sources

Create and use Hadoop HDFS connections in Test Data Manager to use PowerCenter mappings to perform data group movement and data masking operations.

Import metadata from a Hadoop HDFS connection and perform data group movement and data masking operations. A Hadoop HDFS connection type is an HDFS connection that uses PowerCenter mappings to perform a TDM operation. In a TDM plan, you can use a Hadoop HDFS connection as a source, a target, or both.

You can perform the write operation in the following cases:

- Source is a relational database such as Oracle, Microsoft SQL, Sybase, and DB2 for Linux, UNIX, and Windows and the target is a Hadoop HDFS connection type
- Source is a flat file and the target is a Hadoop HDFS connection type
- Both the source and the target are Hadoop HDFS connection types
- Source is a Hadoop HDFS connection type and the target is a flat file

The Hadoop HDFS connection contains the following limitations:

- You cannot perform a write operation when the source is a Hadoop HDFS connection type and the target is a relational database.
- You cannot use shuffle masking when both the source and the target are Hadoop HDFS connection types.
- You cannot use a Hadoop HDFS connection type as a variable when you configure a source or a target.

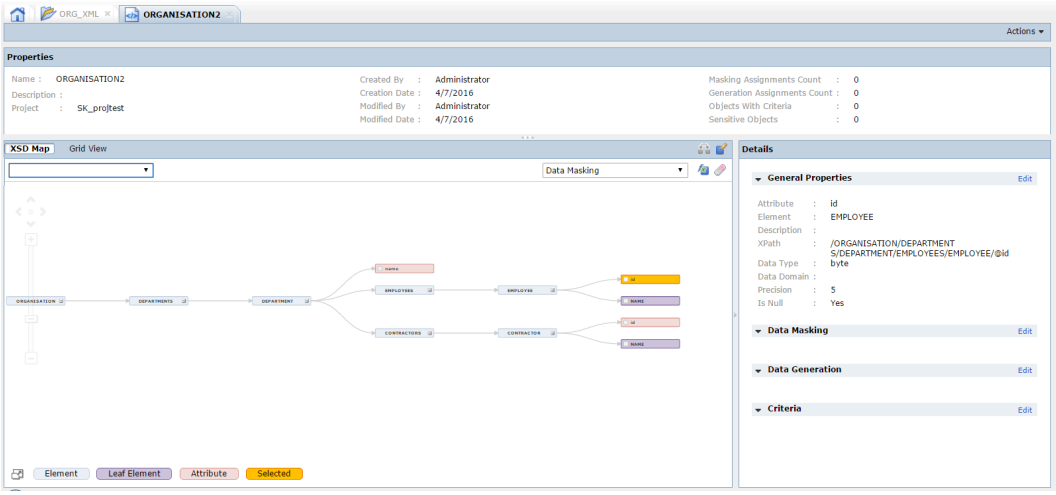
XSD Data Sources

You can perform data subset, data masking, and data generation operations on XML sources.

Import XML source metadata from the PowerCenter repository. To import an XML source, you import the source definition from the PowerCenter repository. A workflow that you generate and run fails if you import the XML instead of the XML source definition (XSD).

You can view the list of XSD files in a project from the **Files** tab in the **Discover** view of the project. Highlight a file to view the properties in the tab below. You can open and view the properties of each file in a separate page.

The following image shows the file page in Test Data Manager:



The file page contains the following tabs:

Properties

Displays the name of the XSD file, project information, and a count of the objects with data masking, data generation, or data subset criteria assignments.

View

You can view the elements in a file as a map or in a grid. View the elements in the file and assign a data domain. To apply a masking rule to multiple elements, you can mark elements as similar value objects. You cannot perform tasks on all elements in the map or grid view. You can choose to view select objects from a list of options in the **Map View**. You can filter and search for elements and attributes based on general properties in the **Grid View**.

Details

Apply subset criteria, data masking rules, or data generation rules from the **Details** tab on the file page. To generate test data, you can select a link between the XML elements and configure the range of child records that you want in the target column.

The **Details** tab has the following tabs:

General

View the properties of an element or attribute. You can add or edit a data domain assignment, and you can mark an attribute or element as restricted or sensitive. When you run a workflow, the PowerCenter Integration Service ignores elements and attributes that you mark as restricted.

If you select a link between the XML elements or attributes, you can configure the minimum and maximum number of child records for each parent record.

Data Masking

Assign a policy and data masking rule or directly assign a data masking rule to an element or attribute.

Data Generation

Assign a data generation rule to an element or attribute, or create and assign a data generation rule.

Criteria

Specify the data subset criteria to apply to an element or attribute.

Editing the Properties of an Element in an XSD File

You can view the general properties of an element or attribute that you select in the map view or grid view. You can edit the properties that are not read-only.

1. Open the project and click the **Files** tab in the **Discover** view.

   A list of XSD files in the project appears.
2. Select the required file to open the file in a separate page.
3. Select the element or attribute that you want to edit.

   The properties of the element or attribute appear on the **Details** tab.
4. To open the **Edit Properties** window, click **Edit** on the **General Properties** tab.

   The **Edit Properties** window lists the following properties:

Property	Description
Name	Read-only. Name of the element or attribute.
XPath	Read-only. The path expression of the element or attribute in the XSD file.
Data Type	Read-only. The data type of the element or attribute.
Precision	Read-only. The precision for the element or attribute. The maximum number of digits or the maximum number of characters that the element or attribute can accommodate.
Data Domain	The data domain assigned to the element or attribute. Select a data domain from the list.
Restricted	The PowerCenter Integration Service does not write the value of a restricted element or attribute to a target. Select this option to mark the element or attribute as restricted.
Sensitive	Indicates whether an element or attribute contains sensitive data. Select this option to mark the element or attribute as sensitive.

5. You can edit the following properties as required:
   - To assign a data domain to the element or attribute, select a data domain from the list of data domains.
   - To mark the attribute or element as restricted, select the **Restricted** check box. During TDM operations the PowerCenter Integration Service ignores elements and attributes that you mark as restricted.
   - To mark the element or attribute as sensitive, select the **Sensitive** check box.
6. Click **OK** to save the changes.

XSD Import Guidelines

Read the following information before you import an XSD file into the PowerCenter repository from PowerCenter Designer:

- To open the **Change XMLViews Creation and Naming Options** window, click **Advanced Options** in the **Import XML Definition** dialog box. Set the following options for the import:
  - Choose to override all infinite lengths with a low value. The value is the default length if the XSD file does not specify a length. You do not require a high value if the elements are of a small length. A high length value requires more memory and affects the performance. Select the option and enter an override value.
  - If the data contains elements that require a higher precision, manually set the precision of the element to the required value after you import the file. Open the file in PowerCenter Designer and select the element in the **Schema Components** panel. Edit the length on the **Actions** tab.
  - Choose to analyze elements and attributes in standalone XML as global declarations. Required in TDM to ensure correct mapping of the namespaces of the different elements and attributes. Import of the XSD in TDM fails if the namespaces of elements and attributes are not mapped correctly.
  - Use the default settings for the other options.
  - Enter a value for the **Default length for anyType element mapped to string** option.
- In the **XML Definition Creation Options** page of the XML Wizard, select **Hierarchy Relationships** and the **Normalized XML Views** option. Use this option to create a root and expand the XML components under the root. When you choose normalized XML views, every element or attribute appears once. One-to-many relationships become separate XML views with keys to relate the views. If you create denormalized XML views, all elements and attributes display in one hierarchical group.

PowerExchange Nonrelational Sources

You can perform TDM operations on nonrelational sources that you define in PowerExchange.

You first need to define datamaps in PowerExchange and import them into PowerCenter. Create the policies, projects, and operations in TDM. Import the PowerCenter sources into a project in TDM.

You can perform TDM operations on the following PowerExchange nonrelational sources:

- Adabas
- IMS unload
- Sequential
- Hierarchical and nonhierarchical VSAM

Note: You cannot import a PowerExchange source directly into the TDM repository.

Importing Data Sources

Import data sources from PowerCenter or from an external database to a project in Test Data Manager.

1. To view a list of the projects in Test Data Manager, click **Projects**.
2. Click a project in the list to open it.
3. Click **Actions** > **Import Metadata**.

   The **Import Metadata** window appears.
4. Choose one of the following options:

- PowerCenter Repository. Import metadata from PowerCenter.
- Datasource Connection. Import metadata from a database connection.

5. Choose whether you want to review the metadata changes before you import the data sources. Test Data Manager displays the information before the import runs. You can choose to skip the import option.

   Test Data Manager shows you the metadata you are importing and the impact on rules and domain assignments or entities and groups in Test Data Manager.

   Test Data Manager shows you the metadata you are importing and the impact on rules and domain assignments or entities and groups in Test Data Manager.

6. If you select an Oracle database connection, you can choose to ignore the tables that do not contain any data.

   Test Data Manager excludes all the empty tables in the Oracle database and lists the tables that contain data.

7. Click **Next**.
8. Choose one of the following options:
   - If you selected **PowerCenter Repository**, select the PowerCenter folder that contains the data source you want to import. You can filter folders by the folder name or the description.
   - If you selected **Datasource Connection**, select the schema to import. You can filter schemas by schema name.
9. Click **Next**.
10. Select the files or tables that you want to import. You can filter the tables by data source, table name, or table description.
11. Click **Next**.
12. Choose when to import the sources. Choose one of the following options:
    - Import Now. Import the data source immediately. To run the import in the background, select **Run import in the background**.
    - Schedule Later. Schedule the import to occur at a specific date and time. Click the calendar to select a date. Use the hour and minute sliders to set the time.
13. Click **Finish**.

    The **Import Progress** dialog box appears with the import progress bar.

14. View the progress of the import job in the **Monitor** view.

    If you chose to import review changes, the **Impact Review** dialog box appears with the metadata, assignments, and project objects.

15. Review the impact, and approve or reject the changes. If you approve the changes, TDM imports the metadata successfully.

    After the job finishes, access the imported metadata through the **Data Sources** details view.

Identity Columns in TDM

Read the following information before you perform TDM operations on identity columns:

- You cannot move data from a source DBO schema to a target non-DBO schema on a Sybase database if the data contains an identity column and indexes.
- If you enable the auto identity option in the database and there are no primary keys or unique keys in a table, Sybase adds the SYB_IDENTITY_COL column. You must mark the column as restricted. A workflow fails if you do not mark the SYB_IDENTITY_COL column as restricted.

- To disable an identity column on a Microsoft SQL Server database, the database user must have execute permission on sp_help.

Delete a Table

You can delete a table from a project if you added too many tables or some tables are not necessary after a period of time.

You can directly delete an orphan table that is not related to another tables and does not contain any assignments. If you delete a table that is a part of profile results, a warning appears and TDM deletes the table from the profile results as well.

You can delete a table that is a part of an entity, a group, or a plan. If the table is part of an entity, a group, a masking plan, or a generation plan, the **Impacted Objects** dialog box appears with the list of affected objects. If the table is part of a template and if the template is part of a plan, the affected objects list does not contain the template and the plan.

You can delete one table at a time. You cannot delete XML files from Test Data Manager after you import metadata from XSD sources.

Deleting a Table

You can delete a table that you added in a data source if you do not use them in a TDM operation.

1. Open a project.
2. Click **Discover** > **Tables**.
3. Select a table that you want to delete.
4. Click **Actions** > **Delete Table**.

   A confirmation message appears if the table does not contain any assignments.
   If the table is part of an entity, a group, a masking plan, or a generation plan, the **Impacted Objects** dialog box appears with the list of affected objects. If the table is part of a template and if the template is part of a plan, the affected objects list does not contain the template and the plan.
5. To confirm and delete the table, click **Continue**.

Project Permission and Security

Use project permissions to control access to projects.

When you create a project, you become the owner of the project. As the project owner, you can add users and user groups and assign the required levels of permission. The domain administrator can also add and edit project permissions.

You can access projects based on the permissions that you have. For example, if you do not have any permissions for a project, it does not appear in the list of projects. You cannot view projects that appear in search results if you do not have the required permissions.

Projects have the following levels of permission:

- Read
- Write
- Execute

Project Permissions

You must have the required project permissions to access and perform tasks in a project.

The following table lists the project permission levels, the tasks that you can perform with each level, and the minimum required privileges for each task:

Permission	Description	Minimum Required Privilege
Read	- Open and view the project. - Monitor logs for the project workflows.	- View project - Monitor project - Audit project
Write	- Open and view the project. - Monitor logs for the project workflows. - Import metadata. - Delete tables. - Create entities, groups, and templates. - Assign rules. - Generate workflows. - Run profiles. - Copy the project. - Delete the project.	- View project - Monitor project - Audit project - Import metadata - Generate project - Manage project - Discover project
Execute	- Open and view the project. - Monitor logs for the project workflows. - Run workflows.	- View project - Monitor project - Audit project - Execute project

Updating User and Group Security

When you create a project, you can assign read, write, and execute permissions to users and user groups. Edit the project permissions assigned to users and user groups from the **Permissions** tab of the project. Changes to permissions take effect from the subsequent login.

1. Open a project and click **Permissions**.

   A list of the users and user groups with permissions for the project appears.
2. Click **Edit** on the **Users** or **User Groups** tab.

   The **Edit Project Permissions** dialog box opens.
3. To edit the permission of a user or user group, select the user or user group from the list and edit the permission as required. You must save the changes for each user or user group.
4. To delete a user or user group, select the user or user group from the list and click **Delete**.
5. To add a user or a user group:
   a. Click **Add Users** or **Add User Groups**.
   b. Select one or more users or user groups.

c. Optional. From the list of permissions, select the required permissions if either of the following statements is true:
   - You selected a single user or user group.
   - You want to assign the same levels of permission to all selected users or user groups.

d. Click **OK**. TDM adds the users or user groups to the list.

e. Select each user or user group and assign the required permission levels. You must save the changes for each user or user group. Skip this step if you performed step c.

6. Click **OK**.

CHAPTER 4

Policies

This chapter includes the following topics:

- Policies Overview, 65
- Policies View, 66
- Policies Task Flow, 66
- Rules, 66
- Data Domains, 67
- Policy Packs, 71
- Import and Export, 74
- Linking Business Glossary Terms to Global Objects , 76
- Policy Management, 77

Policies Overview

A policy is a data masking component that describes the methods to maintain the privacy of specific types of source data.

A policy contains data domains. A data domain describes the functional meaning of a column based on the column data or the column name. For example, a Social_Security data domain contains all database columns with numbers in the following format: 999-99-9999. A Salary data domain might include the Salary, Bonus, and Commission columns in a database.

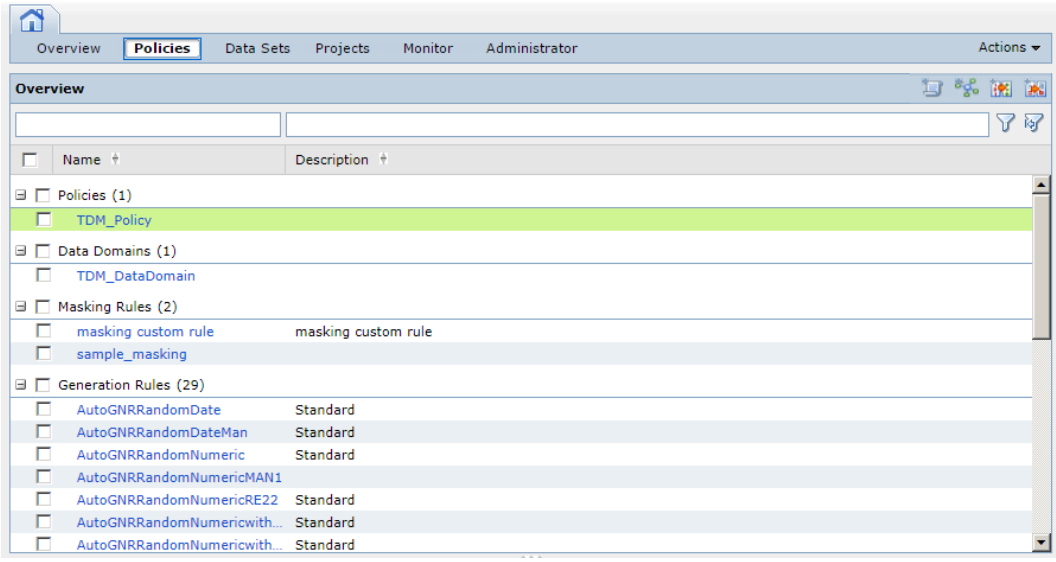
A data domain contains data masking rules. A data masking rule is a data masking technique to mask a specific type of data. For example, you might configure the Substitution data masking technique for First Names and Last Name columns. You configure two Substitution masking rules because each rule contains different parameters.

You can configure data domains, rules, and policies separately. Apply the rules to data domains and add the data domains to a policy. After you define the policy, you can assign the policy to a data source in a project. You can apply a policy to multiple projects.

Policies View

The **Policies** view shows the policies, data domains, and rules in the TDM repository.

The following image shows the **Policies** view in Test Data Manager:



Policies Task Flow

You can create policies, rules, and data domains in any order.

Complete the following high-level steps to define policies:

- Create data masking rules.
- Define data domains to describe which columns can receive the same masking rules.
- Assign the data masking rules to the data domains.
- Create a policy.
- Assign the data domains to the policy.

Rules

A rule defines a logic to mask sensitive data or to generate test data.

Create a data masking rule or a data generation rule and configure the rule parameters. After you add rules to a project, you must assign the rules to columns based on the data types.

You can create rules for string, numeric, and date data types.

You can add the rules to data domains, and use the data domains in policies and plans.

Data Domains

A data domain is an object that represents the functional meaning of a column based on the column data or the column name. Configure data domains to group data source columns for data masking. You can assign a masking rule to a data domain and all the columns in the data domain are masked with the same rule. You can add generation rules to a data domain so that TDM generates data with the same generation rule.

Create data domains to describe the columns you need to mask with the same masking rules. Assign at least one masking rule to each data domain.

For example, you might need to mask all the instances of Social Security number with the same masking rule. You can create a data domain that describes the Social Security data that occurs in the different columns. A database might have a Social Security number in a column called SSN. The database also has a column called SOCIAL_SECURITY in a different table. A Social Security number might also appear in a COMMENTS column.

When you create the data domain, you create a data expression that describes the data format for Social Security numbers. A Social Security number has this format: `999-99-9999`. You can also create multiple metadata expressions that describe possible column names for Social Security numbers. Social Security column names might include `SSN` or `Social`.

You can add data generation rules to a data domain. TDM lists the preferred data generation rules for a data domain. You can edit the list or add another generation rule.

After you define a data domain, you can add the data domain to a policy. You can run profiles for data discovery against data sources in a project. Run profiles to find the columns for data domains. For example, the profile job can find all the Social Security numbers in the source data based on how you defined the data domain. The profile assigns data domains to columns.

Note: If you do not have Data Discovery, you can still use data domains to aggregate data. However, you must manually associate source columns with the data domains.

Apply Masking Rules to a Data Domain

You can assign one or more data masking rules to the data domain. When you assign a masking rule to a data domain, the columns in the domain receive the data masking rule when you configure data masking.

When you assign data masking rules to the data domain, the rules are called preferred rules. If you assign multiple rules to the data domain, you enable one of the rules to be the default rule. The default rule is applied to all columns in the data domain. You can manually change the masking rule for a column to a different preferred rule. You can also apply more than one masking rule to a column.

For example, an organization has a data domain called Last_Name. The Last_Name data domain describes columns that contain last names in company databases. The company can use a shuffle masking rule to mask the last names of customers in a database. The shuffle masking rule is the default rule. The organization applies a substitution masking technique to mask the last names of customers in a different table. The substitution masking rule is a different preferred masking rule in the data domain.

Apply Generation Rules to a Data Domain

You can assign one or more data generation rules to a data domain. When you assign a generation rule to a data domain, the columns in the data domain receive the data generation rule when you configure data generation.

TDM lists the preferred data generation rules for a data domain. You can edit the list or add another generation rule.

When you add data generation rules to a data domain, you can enable one of the rules to be the default rule. The default rule is applicable to all columns in the data domain. To change a default rule, you must edit the data domain and enable the default rule that you want. You can also apply more than one generation rule to a data domain.

For example, an organization has a data domain called Date_of_Birth. The Date_of_Birth data domain describes the columns that contain birth dates of the customers. The organization can use the date random generation rule to generate birth dates. The date random generation rule is the default rule. The organization applies a date sequence generation technique to generate the birth dates of customers in a different table. The date sequence generation rule is a different preferred generation rule in the data domain.

Metadata and Data Patterns for Data Domains

A data pattern and a metadata pattern are regular expressions that you configure to group columns into a data domain. Use regular expressions to find sensitive data such as IDs, telephone numbers, postal codes, and Social Security numbers in the source data.

A regular expression is a text string that describes a search pattern. A regular expression provides a way to match strings of text or patterns of characters in the source data.

A data domain expression can contain data expressions and metadata expressions. A data expression identifies data values in a source. A metadata expression identifies column names in a source. When a data domain contains multiple expressions, any column name or column value that matches an expression in the pattern appear in the search results.

Regular Expression Syntax

A regular expression contains characters that represent source character types, source character sets, and string or word boundaries in the source columns. A regular expression can also contain quantifiers that determine how many times characters can occur in the source data. Regular expressions are case sensitive.

The following special characters are examples of characters that you can include in a regular expression:

Any character except [\^$.|?*+()

All characters except the listed special characters match a single instance of themselves. For example, `abc` always matches `abc`.

\ (backslash) followed by any of the following special characters: `[\^$.|?*+(){}`

A backslash escapes any special character in a regular expression, so the character loses the special meaning.

*** (asterisk)**

Matches the preceding token zero or more times.

[(left bracket)

Marks the beginning of specifications for one character that you want to match.

- (hyphen)

Specifies a range of characters. For example, [a-zA-Z0-9] matches any letter or digit.

] (right bracket)

Marks the end of the specifications for one character.

? (question mark)

Makes the preceding item optional.

{n} where n is an integer > = 1

Repeats the previous item n times.

For information about creating regular expressions, see tutorials and documentation for regular expressions on the internet such as http://www.regular-expressions.info/tutorial.html.

Data Patterns

Data patterns are regular expressions that describe the format of source data in a data domain.

A data pattern can contain multiple data expressions. If any of the expressions match patterns of the data for a column, then the column belongs in the data domain. You can configure detailed regular expressions to identify data in columns.

For example, a Social Security number contains numbers in the following pattern:

```
999-99-9999
```

The following regular expression shows a data pattern that describes the format of a Social Security number:

```
[0-9]{3}-[0-9]{2}-[0-9]{4}
```

Metadata Patterns

A metadata pattern is a regular expression that identifies column names in a source. A metadata pattern can contain multiple metadata expressions.

A metadata expression can be a column name or part of a column name. For example, if you configure `.*Name*` as a metadata expression, column names such as Name, Employee_Name, and Organization_Name in the source appear in the search result.

A column name that matches any metadata expression in the pattern appears in the search results.

A Social Security number might have different column names. The following regular expressions are metadata expression to find Social Security numbers by column name:

```
.*SSN*
.*SOCIAL*
.*SECURITY*
```

Data Domain Options

When you create a data domain you configure options that describe the data domain.

Configure the following options to describe a data domain:

Name

Data domain name.

Sensitivity level

The sensitivity level for all columns in the data domain. The Administrator defines the sensitivity levels that you can choose from when you apply the sensitivity level option.

Description

Description of the data domain.

Status

Data domain status is enabled or disabled. When the data domain is enabled, a profile for data discovery includes the data domain. Default is enabled.

Creating a Data Domain

When you create a data domain, you can enter regular expressions that describe the data that you want to include in the data domain. You can also enter regular expressions that describe the names of database columns to include.

1. To access the policies, click **Policies**.

   The **Policies** view shows a list of the policies, data domains, and rules in the TDM repository.
2. Click **Actions** > **New** > **Data Domain**.
3. Enter the name, sensitivity level, and description for the data domain. Click **Next**.
4. Click **Next**.
5. Optionally, enter a regular expression to filter columns by data pattern.
6. To add more expressions for data patterns, click the **+** icon.
7. To add regular expressions that filter columns by column name, click **Next**. Or, click **Finish** to skip entering any more data domain information.

   You can add multiple expressions.
8. Enter regular expressions to filter columns by column name.
9. Click **Next** if you want to apply preferred masking or generation rules to the data domain. Or, click **Finish** to finish configuring the data domain.
10. To add preferred masking and generation rules to the data domain, click **Add Rules**.

    The **Add Rules** dialog box appears.
11. Select the data masking rules and data generation rules that you want to add.
12. Click **OK**.
13. Enable a default masking rule and a default generation rule.
14. Click **Finish**.

Copying a Data Domain

You can create a data domain by copying a data domain.

1. To access the policies, click **Policies**.
2. Click a data domain description to select the data domain.

   Do not open the data domain.
3. Click **Actions** > **Duplicate**.

   The **Copy <Data Domain Name>** dialog box appears.
4. Change the name and description of the data domain. Click **Save**.

Editing a Data Domain

You can edit a data domain to update the rules, data patterns, and metadata patterns.

1. To access the policies, click **Policies**.

   The **Policies** view shows a list of the policies, data domains, and rules in the TDM repository.
2. Click the name of the data domain that you want to edit.

The data domain opens in a tab.

3. Click **Actions** > **Edit**.

   The **Edit** dialog box appears.

4. To add or edit expressions for data patterns, click the **Data Patterns** tab.
5. To add or edit expressions for metadata patterns, click the **Metadata Patterns** tab.
6. To add or edit masking or generation rules, click the **Preferred Rules** tab. Click **Save**.

   If you delete a rule that contains assignments, the **Impacted Objects** dialog box appears with the list of affected columns and plans.

7. To download the list of affected columns and plans, click **Export**, and save the .csv file.
8. To save the changes, click **Continue**.

   To update the changes in a plan, you must generate and run the plan again.

Deleting a Data Domain

When you delete a data domain, you delete the data domain assignments. When you delete a data domain, you do not delete the rules that you add to the data domain.

1. To access the policies, click **Policies**.

   The **Policies** view shows a list of the policies, data domains, and rules in the TDM repository.

2. Select the name of the data domain that you want to delete.
3. Click **Actions** > **Delete**.

   The **Delete Data Domain** dialog box appears. If you delete a data domain that contains assignments, the **Impacted Objects** dialog box appears with the list of affected columns and plans.

4. To delete the data domain that has no assignments, click **OK**.
5. To delete the data domain that contains assignments, click **Continue**. To download the list of affected objects, click **Export**, and save the .csv file.

   To update the changes in a plan, you must generate and run the plan again.

Policy Packs

A policy pack contains rules that mask common types of sensitive data in the business applications.

TDM includes the following three policy packs:

- PII
- PHI
- PCI

After you install TDM, the policy packs are visible in the **Policies** view. The policy packs include the data and metadata patterns of the data domains. You can view the regular expression strings that define the search patterns in TDM policy packs.

To view and work with policies, you must have the Test Data Manager Service View Policies and Manage Policies privileges. You cannot view the policies if you do not have at least the View Policies privilege.

PII Policy Pack

The Personally Identifiable Information (PII) pack contains data masking rules and policies specific to masking personal information.

The following table describes the data domains and the corresponding default rules available in PII policy pack:

Data Domain Name	Default Rule
Age Domain	Age Rule
Birth Date Domain	Birth Date Rule
Birth Place Domain	Birth Place Rule
Country Domain	Country Rule
Credit Card Domain	Credit Card Numbers Rule
Drivers License Number Domain	Drivers License Rule
Email Address Domain	Email Address Rule
First Name Domain	First Name Rule
Full Address UPPER Domain	Full Address UPPER Rule
Full Name Domain	Full Name Rule
Gender Domain	Gender Rule
Grades Domain	Grades Rule
IP Address Domain	IP Address Rule
Job Position Domain	Job Position Rule
Last Name Domain	Last Name Rule
Organization Name Domain	Organization Name Rule
Passport Domain	Passport Rule
Phone Domain	Phone Number Rule
Salary Domain	Salary Rule
School Name Domain	School Name Rule
Social Security Number Domain	Social Security Number Rule
State Domain	State Rule
Street Name Domain	Address Street Name Rule
UK National Identifier Domain	National Identifier Rule

Data Domain Name	Default Rule
Vehicle Registration Number Domain	Vehicle Registration Rule
ZIP Domain	ZIP Rule

PHI Policy Pack

The Protected Health Information (PHI) pack contains data masking rules and policies specific to the healthcare and the pharmaceutical industries.

The following table describes the data domains and the corresponding default rules available in PHI policy pack:

Policy Pack	Description
Account Number Domain	Account Number Rule
Birth Date Domain	Birth Date Rule
Certificate License Number Domain	Certificate License Number Rule
Device Identifier Serial Number Domain	Device Identifier Serial Number Rule
Email Address Domain	Email Address Rule
First Name Domain	First Name Rule
Health Plan Beneficiary Number Domain	Health Plan Beneficiary Number Rule
IP Address Domain	IP Address Rule
Last Name Domain	Last Name Rule
Medical Record Number Domain	Medical Record Number Rule
Phone Domain	Phone Number Rule
Primary Account Number Domain	Primary Account Number Rule
Social Security Number Domain	Social Security Number Rule
State Domain	State Rule
Unique Identifier Domain	Unique Identifying Number Rule
Web URL Domain	Web URL Rule
ZIP Domain	ZIP Rule

PCI Policy Pack

The Payment Card Industry (PCI) pack contains data masking rules and policies specific to the banking and the finance industries.

The following table describes the data domains and the corresponding default rules available in PCI policy pack:

Policy Pack	Description
Account Number Domain	Account Number Rule
Birth Date Domain	Birth Date Rule
Credit Card Domain	Credit Card Rule
Expiration Date Domain	Expiration Date Rule
First Name Domain	First Name Rule
Last Name Domain	Last Name Rule

Import and Export

You can export policies, domains, and rules to XML files and import the XML files to another TDM repository.

When you export a policy, Test Data Manager exports references to the data domains in the policy. When you export data domains, Test Data Manager exports references to the rules in the repository. However, Test Data Manager does not export the data domains or rules unless you choose to export them.

If a rule or a policy in TDM contains assignments, you cannot import the file. You must either clear the assignment or change the name of the file.

Test Data Manager exports the policies, data domains, and rules to separate XML files. Each file has a default name and number that includes the date. For example: `Policy_121112141309.xml` or `Domain_121112141309.xml`. You can rename the files before you save them.

Exporting Policy Components

Export policies, domains, and rules to separate XML files.

1. In the **Policies** view, choose the policies, data domains, and rules to export.

   To choose an object, select the check box next to the object name.
2. Click **Actions** > **Export**.
3. Click **Open** to open the XML file in a text editor or click **Save** to save the XML file on the system.

   Test Data Manager prompts you for each type of XML file you create. It downloads separate files file policies, data domains, and rules.

Importing Policy Components

You can import policies, data domains, and rules from XML files that have been exported from a TDM repository.

1. In the **Policies** view, click **Actions** > **Import**.
2. Browse for the XML file to import.
3. To import the file, click **Finish** .

   If a policy, data domain, or rule already exists in the TDM repository, Test Data Manager overwrites the original file.

   Note: If a rule or a policy in TDM contains assignments, you cannot import the file. You must either clear the assignment or change the name of the file.

Data Domain Import

Before you import a profile, verify that TDM contains the objects in the profile. So before you import a data domain profile, you must import the data domains in the profile. You can import data domains from the Model repository into TDM. You can edit the data domain in Test Data Manager after you import it.

If you edit a data domain in the Developer tool after you import it, you can import it again to update the data domain in TDM.

You can view data domains that you import in the list of data domains in the **Policies** view.

When you import a data domain from the Model repository, TDM imports the following information in the data domain:

- Name
- Description
- Links to business terms created in the Analyst tool

TDM does not import the following information in the data domain:

- Data pattern
- Metadata pattern
- Preferred rule

If a data domain with the same name exists in TDM, you can choose to overwrite it. If you continue, TDM imports and updates the description and linked business terms of the data domain. It does not overwrite the data patterns, metadata patterns, and preferred rules that you configured within Test Data Manager.

When you import a data domain, TDM merges the linked business terms and retains existing linked business terms. If you re-import a data domain after you delete a linked term from it in the Developer tool, TDM retains the linked term that you deleted. You can manually delete the linked business term from Test Data Manager.

If you re-import a data domain after you change the name of the data domain in the Developer tool or in Test Data Manager, TDM updates the name. You cannot re-import the data domain if a data domain in TDM has the same name.

You can edit data domains that you import in Test Data Manager. You can edit the description, or add a data pattern, metadata pattern, or preferred rule. You can add or delete links to business terms.

Importing a Data Domain

You can import a data domain from the Model repository into the TDM repository. Import data domains before you import a data domain profile.

TDM does not import the data patterns, metadata patterns, or the preferred rules of the data domain.

1. Click the **Policies** view.
2. Click **Actions** > **Import Data Domains**.

   The **Import Data Domains** window opens with a list of data domains in the Model repository.
3. Select the required data domains.
4. Assign a sensitivity level to each data domain that you want to import. Click the **Sensitivity Level** column and choose the required level from the list. Click **Save**.
5. Click **Import**.

   If a data domain with the same name exists, you are prompted to overwrite the data domain or cancel the import. TDM imports the data domain with the description and any linked business terms. You can view the data domain and its properties in the list of data domains.

Linking Business Glossary Terms to Global Objects

You can link business terms in the Informatica Business Glossary to TDM global objects such as policies, data domains, masking rules, and generation rules. A business term includes the business definition and examples of how to use the business term. Use this to store information that users can look up from one central location. When you link a business term to a TDM object, you can view the TDM object when you select the business term in the Analyst tool.

View business terms linked to objects from the **Policies** view. You must configure an Analyst Service in the Test Data Manager Service properties to view or link business terms with TDM objects.

When you select an object from the **Policies** view, a list of business terms linked to the object appears in the Linked Business Terms tab. The list displays the term, the name of the glossary that the term belongs to, and a description.

You can add and remove business term links to objects from the **Policies** view. Click the business term link in the Linked Business Terms tab to view the business term in the Analyst tool.

If you delete a TDM object that has a business term linked to it, you delete the link to the business term. If you edit the name of a linked TDM object, the edit reflects in the glossary. If you delete a business term from a glossary, you delete the object link to the business term.

Linking a Business Term to an Object

You can link business terms to an object on which you have Manage privileges.

Perform the following steps to link a business term to a TDM global object.

1. To access the **Policies** view, click **Policies**.
2. Select the policy, data domain, masking rule, or generation rule to which you want to link a business term.
3. Click **Add** on the **Linked Business Terms** tab. The **Link Asset to Business Term** page opens.

4. Select the glossary from which you want to select a business term. A list of business terms in the glossary appears.
5. Select the terms that you want to link to the object and then click **OK**. The business terms appear in the list of linked business terms.

Deleting a Business Term Link to an Object

You can delete a business term link to objects on which you have Manage privileges. You can delete the link from Test Data Manager or from the Business Glossary UI.

Perform the following steps to delete a business term link to a TDM global object from Test Data Manager.

1. To access the **Policies** view, click **Policies**.
2. Select the policy, data domain, masking rule, or generation rule that you want to edit. The list of business terms linked to the object appears in the **Linked Business Terms** tab.
3. Select the business term that you want to delete.
4. Click **Delete** on the **Linked Business Terms** tab. A confirmation page opens.
5. Click **Yes** to confirm the deletion. TDM deletes the link to the object and updates the **Linked Business Terms** tab.

Policy Management

The **Policies** view shows the policies in the TDM repository. You can create and edit a policy from the **Policies** view.

You can create a policy before or after you define data masking rules and data domains. After you create a policy, you can edit the policy to add data domains and rules. You can copy a policy if you need to create a policy with similar rules or data domains. The duplicate policy contains the rules and data domains from the original policy.

You can export a policy from a TDM repository to an XML file. Import the XML file in another TDM repository.

Creating a Policy

Create a policy in the **Policies** view. You can create the policy before you create data domains and rules. You can add the data domains and rules to the policy at any time.

1. In the **Policies** view, click **Actions** > **New** > **Policy**.

   The **New Policy** dialog box appears.
2. Enter a name and optional description for the policy, and click **Next**.
3. To add data domains to the policy, click **Add Data Domains**.
4. Select the data domains from the list.
5. Click **Finish**.

   The policy appears in the **Policies** view.

Copying a Policy

You can create a policy by copying a policy. When you copy a policy, Test Data Manager copies the data domains and rules in the original policy to the new policy.

1. To access the **Policies** view, click **Policies**.
2. Click a policy description to select the policy.

   Do not open the policy.
3. Click **Actions** > **Duplicate**.

   The **Copy <Policy Name>** dialog box appears.
4. Change the name and description of the policy. Click **Save**.

Editing a Policy

You can edit the policy properties and update the data domains in a policy.

1. To access the **Policies** view, click **Policies**.
2. Click the name of the policy that you want to edit.

   The policy opens in a tab.
3. Click **Actions** > **Edit**.

   The **Edit Policy** dialog box appears.
4. You can change the policy name, description, or status.
5. Click the **Data Domains** tab to edit data domains in the policy.
6. Click **Add** to add data domains to the policy.

   A list of data domains appears.
7. Select the data domains that you want to add to the policy.
8. Click **OK** to select the data domains.
9. Click **Save**.

   If you delete a data domain from the policy and if the data domain contains rule assignments, the **Impacted Objects** dialog box appears with the list of affected columns and plans.
10. To download the list of affected columns and plans, click **Export**, and save the .csv file.
11. To save the changes, click **Continue**.

    To update the changes in a plan, you must generate and run the plan again.

Deleting a Policy

When you delete a policy, you do not delete the data domains or rules that you added to the policy. When you delete a policy, you remove all the assignments that you create with the policy.

1. To access the **Policies** view, click **Policies**.
2. Click to open the policy you want to delete.
3. Click **Actions** > **Delete**.

   The **Delete Policies** dialog box appears.
4. Confirm that you want to delete the policy.

If the policy contains assignments, the **Impacted Objects** dialog box appears with the list of affected columns and plans.

5. To download the list of affected objects, click **Export**, and save the .csv file.
6. To delete the policy, click **Continue**.

   Test Data Manager deletes the policy. To update the changes in a plan, generate and run the plan again.

CHAPTER 5

Data Discovery

This chapter includes the following topics:

- Data Discovery Overview, 80
- Data Discovery Sources, 81
- Discover View, 82
- Column Properties, 83
- Data Discovery Task Flow, 85
- Primary Key Discovery , 86
- Entity Discovery, 87
- Data Domain Discovery, 88
- Column Profile, 90
- Profile Management, 91
- Profile Import, 95
- Apply the Results, 96
- Project Tables, 100

Data Discovery Overview

Run profiles to discover source data for data masking and data subset operations.

Before you run data masking, you can discover which columns to mask with the same masking rules. Before you create a data subset, you can discover relationships between tables and the entities for the data subset database. You can apply profiling results instead of manually configuring data subset entities or manually applying a data masking rule to one column at a time.

You can run the following types of profiles:

Foreign key

Determines relationships between tables. When you run the profile, the Data Integration Service evaluates the structure of the source tables and identifies potential table relationships. Select entities from the profile results. TDM adds the logical constraints in the TDM repository, but does not apply these constraints in the source tables. When you generate a workflow from a data subset plan, TDM uses the constraints to create subset data with referential integrity.

Primary key

When you run the profile, the Data Integration Service evaluates the structure of the source tables and identifies potential primary keys. The primary key profile includes the percentage of unique values in a column or combination of columns. Select primary keys from the profile results. TDM adds the primary keys in the TDM repository, but does not add them to the source tables.

Data domain

Identifies the columns that belong in a data domain based on the data value of the column or the column name. Use the results when you apply data masking rules. You can apply a rule to all columns that belong to the data domain instead of manually applying the rule to one column at a time.

Column

Analyzes data quality in selected columns in a table. A column profile determines the characteristics of columns in a data source, such as range of values, list of values, and data patterns. When you create a column profile, select the tables on which you want to run a profile and configure data sampling options.

You can run data domain profiles and primary key profiles at the same time. You can also import and run profiles that you create in Informatica Developer.

Data Discovery Sources

You can run profiles for data discovery on relational sources.

You can run profiles for data discovery on the following sources:

- Oracle
- Microsoft SQL Server
- IBM DB2 for Linux, UNIX and Windows
- MySQL
- Sybase
- Teradata

Use ODBC to connect to MySQL, Sybase, and Teradata sources.

Rules and Guidelines for Data Discovery Sources

Use the following rules and guidelines when you configure the data discovery sources:

- You cannot run a profile for data discovery on nonrelational sources. However, you can use profiling in PowerCenter or Informatica Data Quality to create and run profiles and then export the results. You can import the results to the TDM repository with Test Data Manager. For information on how to run profiles in Informatica Data Quality and export them, see the Informatica *Profile Guide*.
- You can import the results of profiles that use specific options in Informatica Data Quality.
  - You can view the results of profiles that use the Enterprise Discovery Profile and Profile options.
  - You can import the results of data discovery performed with mapplets created using simple regular expressions. You cannot import results if the mapplets use labeler, tokenizer, content set, or reference tables.
  - You can import data domains from the Model repository into TDM. TDM does not import the data pattern, metadata pattern, and preferred rules.

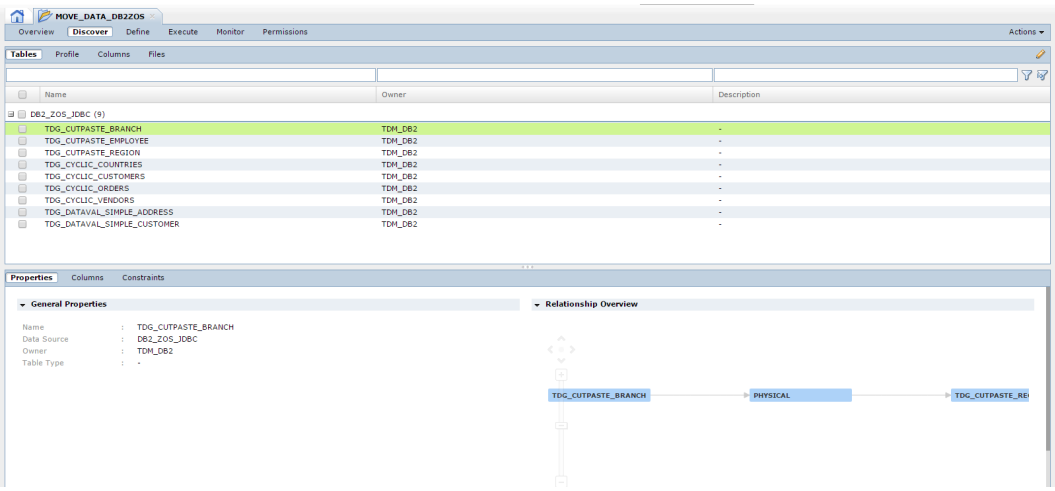
- You can import and view domain discovery results of profiles run in the project. You can import the results of profiles created in folders within the project, but you cannot view the results in TDM.

- A table that you import to the repository must have the same connection when you use it in a profile. If you use a different connection for the profile than you used to import the data source, you might receive unexpected results.
- You cannot run a profile that contains two tables with the same name. For example, a project might have more than one EMPLOYEE table. Each EMPLOYEE table might have a different owner. You must create a separate profile for each EMPLOYEE table.
- When you import source metadata of a MySQL source from the PowerCenter repository, PowerCenter converts the unique key constraints defined in the MySQL tables to primary key constraints. You see the unique key constraints defined in the MySQL tables as primary keys in the Columns tab on the **Discover | Tables** page.

Discover View

View tables, manage primary key-foreign key relationships, configure profiles, and update columns for masking in the **Discover** view.

The following image shows the **Discover** view in a Project:



The **Discover** view contains the following views:

Tables

View the tables in the data source. Select a table and view the relationship between the table and other tables in the project. View the columns in a table. Define primary key and foreign keys in tables. You can delete a table from the metadata.

You can disable physical primary key columns when you do not need the composite primary keys or entity relationships. In the **Tables | Columns** view, you select a row with a physical primary key and enable or disable the primary key from the **Actions** menu. You can also edit the precision of a source table column based on the values that you need in the target column.

Profile

Define a profile for data discovery. Configure the profile to find columns for data domains and possible primary key-foreign key relationships between tables.

Columns

Maintain data domain assignments. Mark columns as restricted, or sensitive, assign bulk data domains, and define similar column values for value cascades.

Files

View a list of XSD files in the project. Select a file in the list to open the file information in a separate tab. You can perform data subset, data masking, and data generation tasks on XSD sources.

Column Properties

You can configure column properties for a project source.

The following table describes the properties listed on the **Discover | Columns** view:

Property	Description
Table Name	Name of the table.
Column	Name of the column.
Owner	Name of the database schema.
Data Type	Column data type.
Domain	Name of the data domain that the column belongs to.
Restricted	The column in the database is a restricted column. The PowerCenter Integration Service does not write the value of a restricted column to a target.
Computed	Indicates whether a column contains data that is computed from other columns. You cannot assign rules to a computed column.
Sensitive	Indicates whether a column contains sensitive data.
Similar Value Columns	Shows the number of columns that are related to this column in a value cascade.

Sensitive Columns

When you mark a column as sensitive, you indicate that the column contains sensitive data. Sensitive columns are for reference only and they do not affect processing.

When you assign a data domain to a column, Test Data Manager marks the column as sensitive.

Computed Columns

Computed columns are columns that contain data determined by data in other columns.

For example, an *Employee_Salary* table contains columns *Emp_ID*, *Basic_Salary*, *Allowances*, *Tax_Deduction*, and *Net_Pay*.

Net_Pay is a computed column that is calculated as `Basic_Salary + Allowances - Tax_Deduction`. The database populates the data in *Net_Pay*. If the values in any of the three columns change, the database updates the value in the *Net_Pay* column.

You can include tables that contain computed columns in a TDM operation. Because the values are calculated based on data in other columns, the TDM operation does not copy data into computed columns in the target. The database populates the values based on the data in other columns.

You cannot apply data masking or data generation rules on computed columns. You cannot create a value cascade or an auto cascade on computed columns.

You cannot perform data generation operations that include an entity if the primary key column is a computed column.

Value Cascades

Use a value cascade to mask similar columns in different tables with the same masking rules. Use a value cascade when the columns that you want to mask are not related by key constraints. Data masking returns the same values into each column. Configure a group of similar value columns to define a value cascade.

Configure value cascades as similar value columns and define one of the columns as the driving column. If you assign a rule to the driving column, Test Data Manager assigns the rule to all the columns in the group. For example, you configure employee_num column in one table and EmpID in another table as similar value columns. Define employee_num as the driving table. When you assign a data masking rule to employee_num, EmpID receives the same masking rule. The columns in a value cascade must be the same type.

When you view the columns in a project, a driving table shows a number in the **Similar Value Columns** column. The number indicates how many other columns are in the value cascade. The similar value columns do not show a number in the **Similar Value Columns** column.

You can view a list of the columns in a value cascade. Select the driving column in the content panel. Click the **Similar Value Columns** view in the details panel. A list of the similar value columns appears.

To configure value cascades for an XSD source, open the XSD file from the **Discover | Files** view within a project. Select the elements and attributes, and configure similar value columns.

Creating a Value Cascade

Configure a value cascade as a group of similar value columns and define one of the columns as the driving column.

1. In the **Discover | Columns** view, select multiple columns that you need to define as similar columns.
2. Click **Actions** > **Set as similar value columns**.

   A list of the columns with similar values appears.
3. Choose one column as the driving column.

   To select the column, click the driving table column checkbox to change the value from No to Yes.

Changing a Value Cascade

You can change a value cascade configuration. You can change the driving column, delete a column, or add similar value columns.

1. In the **Discover | Columns** view, select the driving table of the value cascade you want to change.

   A driving table has a numeric value in the **Similar Value Columns** column.

2. Click the **Similar Value Columns** view in the **Details** panel.

   A list of the columns in the value cascade appears.

3. Click **Edit Similar Value Columns**.

   The **Edit Similar Value Columns** dialog box appears.

4. To change the driving column to a different column, change the **Driving Column** value to Yes for the column that you want.

   You do not have to disable the previous driving table column. The value changes to No by default.

5. To add similar value columns to the group, click **Add**.

   Select the columns from the **Add Similar Value Columns** dialog box.

6. To remove columns from the group, select the columns on the **Edit Similar Values** dialog box and then click **Delete**.

Deleting a Value Cascade Group

You can delete a value cascade.

1. In the **Discover | Columns** view, select the driving table for the value cascade.

   A driving table has a numeric value in the **Similar Value Columns** column.

2. Click the **Similar Value Columns** view in the **Details** panel.

   A list of the similar value columns in the value cascade appears.

3. Select any column in the group and click **Delete.**
4. Confirm that you want to delete the similar values group.

   Test Data Manager removes the value cascade.

Auto Cascades

An auto cascade masks the primary key in a parent table and related foreign keys in child tables with the same value.

Tables with key constraints have auto cascades applied by default. You can manually disable or enable auto cascades. For example, you need to mask the employee ID in an employee master table and two child tables. The employee ID is a primary key in the master table. When you mask the employee ID in the employee table, the employee ID in the child tables receives the same masked value.

To use auto cascades, key constraints must exist between the parent and the child tables.

You cannot use auto cascades for XSD, MongoDB, and Cassandra data sources.

Data Discovery Task Flow

You can run profiles for data discovery to find primary keys and entity relationships between tables in the source data. You can run a data domain profile to search for columns to assign to data domains for data masking.

Before you can run profiles, the administrator must configure a connection to the source database for data discovery. The administrator must also configure connections to the Data Integration Service and the Model Repository Service.

Complete the following high-level steps to perform data discovery:

1. Create a profile.
2. Select the type of profiling you want to perform. You can choose to run a primary key profile, an entity profile, or a data domain discovery profile.
3. If you choose to run a data domain discovery profile, choose the data domains to search for.
4. Choose the sampling size for the profile.
5. Run the profile and monitor the job.
6. After the job completes, open the profile again.
7. Review the primary key profile results, the entity profile results, and data domain profile results.
8. Select and approve the results that you want to use for data masking and data subset operations.

Primary Key Discovery

To discover possible keys in source tables, create a primary key profile.

To identify potential data quality issues, create a primary key profile that discovers non-unique rows. A primary key discovery profile is also useful when the primary key is a combination of columns that uniquely defines a row. Non-conforming rows might contain duplicate information. After you discover potential keys in a table, you can add the keys to the source table in a project.

For example, an Employee table contains an EmpID column. The value of the column is 100% unique. A primary key profile identifies the column as a possible primary key that you can add to the table in the project.

Primary Key Profile Options

When you run a primary key profile, the Data Integration Service infers primary key candidates from the source tables. To limit the candidate results, configure inference options.

A primary key profile identifies the columns and combinations of columns that meet a specific conformance level. You can edit the maximum number of columns to combine for unique key identification.

The following table describes the options for primary key profiles:

Option	Description
Primary Key Rows	The maximum number of columns that the profile can combine to form a primary key. Default is 3.
Max Rows	The maximum number of rows to profile. Default is 1,000.
Conformance Criteria	The minimum percentage or maximum number of key violations that the profile allows. Default is a minimum percentage of 99.

Entity Discovery

To discover data relationships between source tables, create an entity profile. Entity discovery shows the tables that are potential entities.

The entity profile discovers entity relationships in the source data. An entity consists of related parent and child tables. The entity profile returns a list of possible entities by comparing column names in the source tables. You can select entities from the profile results to add to the project. When you run a data subset operation, the entities that you define for the subset determine the subset database structure and the data in the subset.

For example, the Employee table, the Employee_Detail table, and an Employee_Address table contain an EmpID column. The entity profile suggests that the Employee table could be an entity with two direct relationships. The Employee table has two child tables. When you view the profile results, you select the Employee table as an entity. When you run a data subset operation, you can create a filter on EmpID to retrieve the data for employees with an EmpID less than 2000. The data subset operation filters data for each table in the entity relationship.

If you select Entity as the profile type, and the physical database does not contain primary keys, the profile does not return any results. If you run the profile as a primary key and entity profile, the profile discovers possible primary keys and foreign keys. The profile returns results even if the physical database does not contain primary keys.

Entity Profile Options

TDM infers foreign key candidates from the source tables. To limit the candidate results, specify inference options.

Define options to configure how the entity profile matches the datatypes of the primary keys and the foreign keys of the entity table.

The following table describes the entity profile options:

Option	Description
Datatypes used in comparisons	The datatype to use for comparisons. Select one of the following options: - Inferred Datatypes. Compares the primary key and the foreign key based on the datatypes inferred by TDM. - Documented Datatypes. Compares the primary key and the foreign key based on the datatypes defined for the source column.
Comparison case sensitivity	Performs case-sensitive comparisons.
Trim spaces before comparison	Trims spaces before comparing values. Select one of the following options: - None. Does not trim leading or trailing spaces. - Both. Trims both leading and trailing spaces. - Leading. Trims leading spaces. - Trailing. Trims trailing spaces.
Maximum foreign keys returned	Maximum number of matched foreign key results that you want to return.

Option	Description
Minimum confidence percent	Minimum percentage of foreign key values that match the inferred datatype.
Regenerate signature	Determines whether the data profiling warehouse should clear the existing object information before the Data Integration Service runs the profile.

Data Domain Discovery

Data domain discovery finds the source columns that contain similar data. The profile assigns the same data domain name to each column that contains similar data. You can assign the same data masking rules to all the columns in a data domain at the same time.

Create a data domain to describe the columns you need to mask with the same data masking rules. When you create a data domain, you configure regular expressions that define patterns in the data or patterns in the column names.

Run the data domain discovery profile to find the columns that match the criteria in the data domain regular expressions. When you configure a profile for data domain discovery, select the tables to search in the data domain discovery operation. Select which data domains to search for in the tables. You can select policies that contain data domains instead of selecting each data domain to search for.

After you run the profile for data discovery you can view the profile results. The profile results assign source columns to data domains. You can choose which profile results to use for data masking.

Data Domain Profile Sampling Options

When you run a profile for data domain discovery, configure sampling options to limit the number of rows to search or limit the regular expressions to search with.

The following table describes the sampling options that you can select in a profile for data discovery:

Option	Description
Data	Search for patterns in the data only.
Column Name	Search for patterns in the column name only.
Data and Column Name	Search for patterns in the data and in the column name.
Maximum Rows to Profile	Limit the number of rows to profile. Default is 1000.
Minimum Conformance Percent	Minimum percentage of rows where the column data or metadata matches the data domain.

Assigning a Data Domain to Multiple Columns

You can manually assign a data domain to multiple columns at a time. You can also remove the data domain assignment from multiple columns at a time.

1. Open a project.
2. Navigate to the **Discover | Columns** view.

   A list of all the columns in the project appears.
3. Select the columns that you want to assign the data domain to.
4. Click **Actions** > **Edit Assignments**.

   The **Edit Data Domain Assignment** dialog box appears.
5. Choose the data domain to assign to the columns.

   You can choose a blank data domain to remove the previous data domain assignment.
6. Click **Save**.

   The data domain assignments appear in the **Columns** view.

Manually Updating the Column Data Domain

You can manually update a data domain for a column. When you add a data domain to a column, Test Data Manager marks the column as a sensitive column.

1. Open the project and click **Discover | Columns**.
2. Click the **Domain** column for the column you want to update.
   A list of data domains appears.
3. Select the data domain to add to the column.

Note: You can run a profile for data domain discovery to update the data domain for columns.

Data Domain Assignments Import and Export

You can import a data domain assignments file into Test Data Manager and export data domain assignments from Test Data Manager.

You can export a data domain assignments file from Secure@Source and import the data domain assignments file into Test Data Manager.

Import a data domain assignments file in CSV format and view the data domain assignments in Test Data Manager. If a data domain assignment with the same name exists, TDM overwrites the existing data domain assignment with the imported data domain assignment. Any error during import is logged in the log file.

When you import a data domain assignments file, the records from the file are imported sequentially. If TDM fails to import a record, an error is logged and the import resumes from the next record. You can view the errors in the **Administrator | Application Logs** tab.

The CSV file that you import must contain columns in the following order:

- Data Source Name
- Owner Name
- Table Name
- Column Name

- Is Sensitive
- Domain Name

TDM considers the first row of the file as the header and does not import the first row. If the CSV file does not contain a header, you lose the first row of data.

Export data domain assignments as CSV files. You can import the CSV file into Test Data Manager.

To import or export a data domain assignments file, you must have read and write permissions on the project and Discover Project privilege.

Exporting a Data Domain Assignments File

1. Open the required project and click the **Discover | Columns** view.
2. Click **Actions** > **Export Data Domain Assignments**.
3. If you configure the browser to prompt for a download location, you must enter a file path and file name.

   Click **Save** to save the file.

Importing a Data Domain Assignments File

1. Open the required project and click the **Discover | Columns** view.
2. Click **Actions** > **Import Data Domain Assignments**.

   The **Import a Data Domain Assignments File** dialog box appears.
3. Select the CSV file that you want to import and click **Finish**.

   The imported data domain column assignments appears in the **Discover | Columns** view.

Column Profile

Create and run a column profile to determine the data source column characteristics, such as a list of values, data patterns, and value ranges. Create a column profile, and extrapolate the profile results to generate test data.

Select the columns on which you want to run a profile and configure data sampling options when you create a column profile.

After you configure and run a column profile, TDM analyzes column data and infers rules based on the data. You can view the profile results for the columns. You can manually assign the inferred rules to the columns in the profile results or apply the inferred rules that TDM provides.

You can run column profiles on Oracle and Microsoft SQL Server databases.

Inferred Rules

After you configure and run a column profile, TDM infers analyzes column data and infers rules based on the data. Column data types include all the inferred data types for each column in the profile results.

Column profile results contain the following types of inferred rules:

Range

The range of column values, such as the maximum and minimum lengths of values and the first and last values in each column. The range of values includes the percentage of null values in the columns. You can use rang of values for numeric and date data types.

Patterns

The patterns of data in each column and the percentage of distribution for each pattern. The sum of the distribution percentage must be equal to 100%. You can use data patterns for string and numeric data types.

List of Values

The list of unique and nonunique values in the columns and the percentage of distribution. When you assign the inferred rule type to generate test data, the sum of the distribution percentage must be equal to 100%. You can use list of values for string, numeric, and date data types.

You can assign the inferred rule type to the target columns from the **Define | Data Generation** page. Auto rules are the inferred rules that you assign after you run a column profile. When the status of inferred rule is Yes, you can assign Auto Rule to target columns to generate test data. You can edit the properties of the inferred rule types from the **Define | Data Generation** page.

Column Profile Options

You can configure the sampling options when you create or edit a column profile.

The following table describes the sampling options for a column profile:

Option	Description
Minimum conformance percentage for patterns	Required. The minimum percentage of patterns that you want in the profile results. Default is 5. If the patterns are less than the minimum conformance percentage, TDM does not display the patterns in the profile results.
Maximum number of rows to be profiled	Required. The maximum number of rows that you want to run the profile on. You can choose all rows in the data object or enter the number of rows that you want to profile on. Default is All.
Minimum data coverage to show List of Values	Required. The minimum percent of data that you want in the list of values. Default is 80. If the maximum number of values that you want in the list of values is 20, then 80% of the data must contain these 20 values.
Maximum number of values in List of Values	Required. The maximum number of values that you want in the list of values. Default is 20.

Profile Management

You can create and run profiles, add primary and unique keys to the source, and verify and add constraints to the source. Alternatively, you can import profiles that you create and run in Informatica Developer.

You can run a profile multiple times, and you can edit a profile between runs.

Creating a Primary Key Profile

Create a primary key profile to identify a list of possible primary keys that you can add to tables without keys. Create a primary key profile if the source data does not contain primary and foreign key relationships.

1. Open the project, and click the **Discover** view.
2. Click the **Profile** view.
3. Click **Actions** > **New Profile** to create a profile.
4. In the **New Profile** dialog box, enter the profile name and description. Choose to create a primary key profile.
5. Click **Next.**
6. Select the tables to profile, and click **OK**.
7. Click **Next.**
8. In the **Sampling Options** page, enter the maximum number of columns that you want to include in the primary key.
9. Enter the maximum number of rows that you want to run the profile on.
10. Enter the minimum conformance required, or enter the maximum number of rows that can violate the criteria.
11. Click **Finish**.
12. Click **Actions** > **Execute**.

Creating an Entity Profile

The entity profile creates a list of possible parent tables that you can define for data subset operations. Create and run entity profiles in the **Discover** view.

1. Open the project, and click the **Discover** view.
2. Click the **Profile** view.

   The **Profile** view shows a list of the profiles in the project.
3. Click **Actions** > **New Profile** to create a profile.
4. In the **New Profile** dialog box, enter the profile name and description. Choose to create an entity profile.

   Select primary key and entity profile if the physical database does not contain any primary keys.
5. Click **Next**.
6. Select the tables to profile, and click **OK**.
7. Click **Next**.
8. In the **Sampling Options** page, enter the maximum number of columns that you want to include in the primary key.
9. Enter the maximum number of rows that you want to run the profile on.
10. Enter the minimum conformance required, or enter the maximum number of rows that can violate the criteria.

The following table describes the entity options that you can update:

Entity Option	Description
Data types used in comparison	Determines how Test Data Manager derives data types to compare in the tables. Choose one of the following options: - Inferred data types. Compares the primary key and the foreign key based on the data types inferred by TDM. - Documented data types. Compares the primary key and the foreign key based on the data types defined for the source column.
Comparison case sensitivity	Indicates if the foreign key comparison is case-sensitive or not. If you choose case-sensitivity, profile performance might be affected.
Trim values before comparison	Trims leading values, trailing values, or leading and trailing values.
Minimum conformance percent	Minimum percentage of rows without key violations that the profile allows for foreign key discovery.
Regenerate Signature	Determines whether the data profiling warehouse should clear the existing object information before the Data Integration Service runs the profile.

11. Click **Save**.
12. Click **Actions** > **Execute** to run the profile.

Creating a Data Domain Profile

Create and run data domain profiles in the **Discover** view.

A project must contain policies before you create a data domain profile. The policies contain the data domains that you can use in a profile for data discovery.

1. Open the project and click the **Discover** view.
2. Click the **Profile** view.

   The **Profile** view shows a list of the profiles in the project.
3. Click **Actions** > **New Profile** to create a new profile.
4. In the **New Profile** dialog box, enter the profile name and description. Choose to create a data domain profile.
5. Select the tables to profile and click **OK**.
6. Click **Next.**
7. In the **Select Sampling Options** pane, choose whether to add policies or data domains to the profile. When you select a policy, Test Data Manager includes all data domains in the policy.

   Test Data Manager returns a list of policies or data domains in the pane.
8. Select the policies or the data domains to profile.
9. In the **Sampling** panel, select whether to run data discovery on the source data, the column name, or the data and the column name.

   You can run a profile for column metadata and then run it again for the source data.
10. Enter the maximum number of rows to profile.
11. Enter the minimum conformance percent.

All rows might not conform to the data domain expression pattern. You can enter a minimum percentage of the profiled rows that must conform.

12. Click **Save.**
13. Click **Actions** > **Execute**.

Creating a Column Profile

To perform a data generation operation, you can create and run a column profile.

1. Open a project and click the **Discover** view.
2. Click the **Profile** view.

   The **Profile** view shows a list of the profiles in the project.
3. Click **Actions** > **New Profile**.

   The **New Profile** dialog box appears.
4. Enter the profile name and description.
5. Select **Column**.
6. Click **Next**.
7. To add the tables that you want to run a profile on, click **Add**.

   The **Select Tables** dialog box appears.
8. Select the tables, and click **OK**.

   The list of selected tables appears.
9. Click **Next**.
10. To create the column profile, configure the required sampling options.
11. Click **Finish**.

    The profile opens in a tab.
12. To run the profile, click **Actions** > **Execute**.

    The **Execute Profile** dialog box appears.
13. Select a connection to run the column profile.

    Note: A column profile run fails if you select a native Oracle ODBC connection. Create an Oracle ODBC connection with the DataDirect driver.

    A column profile run fails if you do not specify a schema name in the Microsoft SQL Server database connection.
14. Click **Execute**, and close the profile.
15. In the project, click **Monitor** to view the progress of the profile run.

Editing a Profile

You can edit a profile and run it multiple times. When you run the profile, the Data Integration Service overwrites the original profile results.

When you edit a profile, you can change the profile name and description. You can add or remove tables. When you run data domain profiles, you can change the policies and data domains in the profile. When you run primary key profiles and entity profiles, you can change the options.

1. Open the project and click the **Discover** | **Profile** view.

2. In the **Discover** view, click a profile name to open it.
3. Click **Edit**.

   The **Edit Profile** dialog box appears. The dialog box has the **General** tab, the **Tables** tab, and the **Relationship Sampling** tab or the **Sampling** tab depending on the profile type.
4. On the **General** tab, you can edit the name and description of the profile.

   You cannot change the profile type.
5. On the **Tables** tab, add or remove tables that you want to profile.

   To add a table, click **Add** and select the tables you want to add. To delete a table, select the check box for the table you want to delete. Click **Delete**.
6. On the **Relationship Sampling** tab, select to override default inference options. Edit the options for finding primary key and foreign keys in the source data.
7. On the **Sampling** tab, you can change the data domains or policies in the profile. You can choose to run the profile against the source data, the column name, or both data and column name.
8. Click **Actions** > **Execute** to run the profile.

Deleting a Profile

You can delete profiles.

1. Open the project and click the **Discover** view.
2. In the **Discover** pane, select the profile row, but do not open the profile.
3. Click **Actions** > **Delete** to delete the profile.
4. Click **Yes** to confirm the delete.

Profile Import

You can import a profile from the Model repository and view the profile in Test Data Manager.

For example, you can run a profile on a flat file source in the Developer tool. You can import the profile into TDM. You can then use the profile information to apply masking rules to mask source data or to create a data subset in TDM.

Import the profile from within a project. You can run a single profile to identify multiple types of data. So a profile might contain data domain, primary key, foreign key, and column profile information. Select the types of profile information that you want to import.

Before you import a profile, import the metadata and the objects in the profile into the TDM repository. For example, before you import a data domain profile, the TDM repository must contain all the data domains that appear in the profile. You can import the data domains from the Model repository. You cannot view the profile after import if data domains with the same name do not exist in the TDM repository.

You can import profiles with table names that contain alphanumeric characters or the special character _.

Importing a Profile

You can import profiles from the Model repository into TDM. You can then view the profile in Test Data Manager.

Verify that the source metadata and the data domains are in the TDM repository before you import the profile. Import the metadata into the project from a connection by the same name as the connection that you used to run the profile

If you delete the metadata from the project after you import a profile, the profile does not display results. If you import the metadata again, and import the profile again with a different name, the results appear in the first profile as well.

1. Open the project and click the **Discover** | **Profile** view.
2. Click **Actions** > **Import Profile**.

   The **Import Profiles** dialog box appears.
3. Select the profile that you want to import from the Model repository. Click **Next**.

   You cannot import a profile that contains a table name with special characters other than _.
4. Optional. TDM imports the profile with the same name and description. Edit the profile name and description if required.
5. Optional. Choose to import the curation information with the profile.

   When you import the curation information, TDM imports the saved review information and performs tasks based on the review information. For example, TDM assigns a data domain to a column if the inferred data domain is approved in a data domain profile.
6. Select the required profile types.

   By default, TDM selects all the profile types that the profile contains. You can choose the profile types that you want to import.
7. If you select the **Column** profile type, enter the column profile properties.
8. Click **Finish**.

   The imported profile appears in the **Discover** view.

Apply the Results

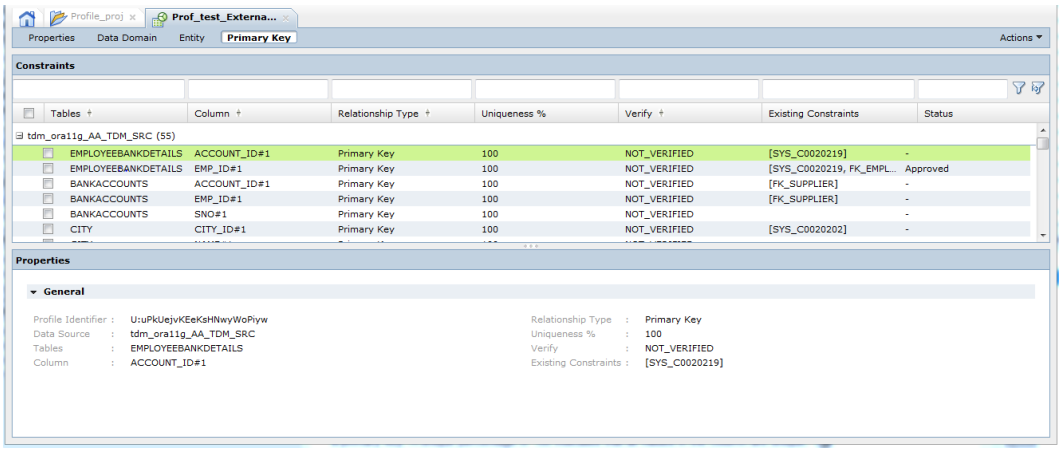
After the profile run, you must close the profile in Test Data Manager and open it again to view the profile results.

Click separate views for the primary key results, the entity profile results, the data domain profile results, and the column profile results.

Primary Key Results

The primary key results is a list of possible primary keys that you can add to tables that do not have keys. Use the unique percentage value in the profile results to determine which column or combination of columns that you might add as a primary key for a table.

The following image shows sample primary key profile results:



To view primary key results, close the profile and open it again to display the results. Click the **Profile | Primary Key** view. Examine the primary key profile results. The primary key candidates include the percentage of unique values in a column or combination of columns.

The following table describes the fields in the primary key results:

Property	Description
Table Name	The name of the table that contains the column with the primary key candidate.
Column Name	The name of the column or combination of columns that the Data Integration Service infers to be a primary or unique key.
Relationship	The function of the column in the table is primary key.
Uniqueness	The percentage of unique values in the column. Use the unique percentage value in the profile results to determine which column or combination of columns to add as a primary key. A unique percentage of 100 indicates that all values in the column are unique.
Verified	Indicates whether the primary key candidate has been verified as not containing another constraint.
Existing constraints	Shows any existing primary key or foreign key constraint for the column.
Status	Indicates whether a user verified, approved, or rejected the primary key candidate.

Apply the Primary Key Profile Results

You can choose primary key candidates from the profile results and add the keys to tables in the project.

You can configure a status that indicates whether you want to accept the primary key candidate or reject the primary key candidate. When you approve a primary key, the Test Data Manager adds the key to the table in the TDM repository.

Select a column and change the status from the **Actions** menu. Select **Verify**, **Approve**, or **Reject**.

When you approve the primary key candidate, the Test Data Manager adds the primary key to the table. When you reject the primary key candidate, it does not affect the subset operation.

Use the **Verified** column to track results as you finish the review. The **Verified** column does not affect data subset operations.

Entity Discovery Results

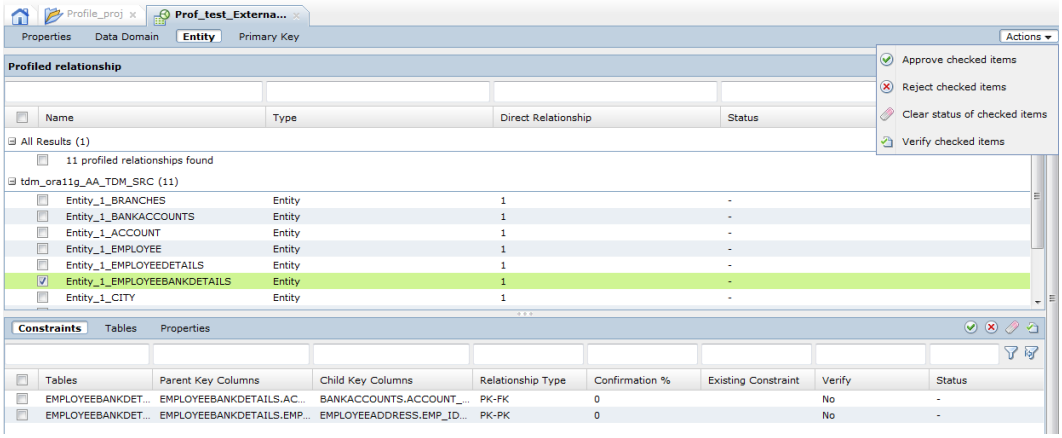
The **Entity** view shows the list of possible entity relationships in the source data. You can select a column in the profile results and see other columns that might contain the same data. You can select entities from the profile results to add to a data subset operation.

To view entity discovery results after you run the profile, close the profile and open it again. Click the **Profile | Entity** view.

Select an entity to view the constraints in the **Constraints** view. Test Data Manager shows the parent columns, the child columns, and the relationships. Approve the constraints to add them to an entity.

You can approve each entity that you want to add to a project.

The following image shows the results of an entity discovery profile:



Data Domain Discovery Results

The data domain profile results show a list of source columns and possible data domains to assign to the columns. You can select which data domain candidates to use for data masking from the profile results.

To view data domain discovery results, close the profile and open it again. Click the **Profile | Data Domain** view.

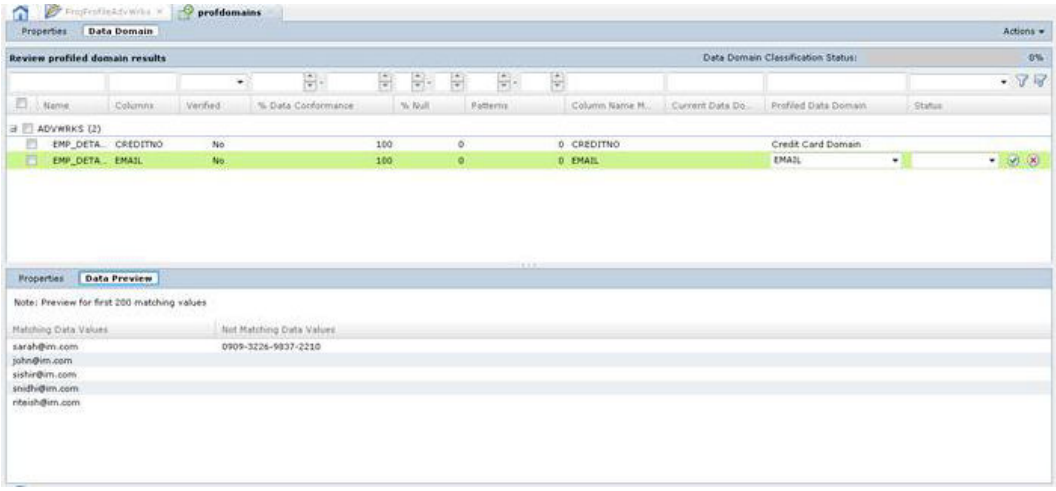
Select a column and click on the **Data Preview** tab to view the source data of the selected column. The data viewer displays the first 200 records of columns returned in the data domain profile.

You can select rows and approve the data domain for each column. When you approve the suggested data domain for the column, you can assign the rules in the domain to each column in the data domain. You can assign the rules in the **Define | Data Masking** view.

After you are finished working on the data domain for a column, you can verify the data domain for each column in the **Data Domain** view. The **Verify** column is for tracking. It does not affect the data domain profile operation.

When you finish approving the data domains, you can mark the data domain classification as completed. Use this method to verify that you reviewed all the results. Completing the data domain classification does not affect any process.

The following image shows the data domain discovery results:



The following table describes the columns in the data discovery results:

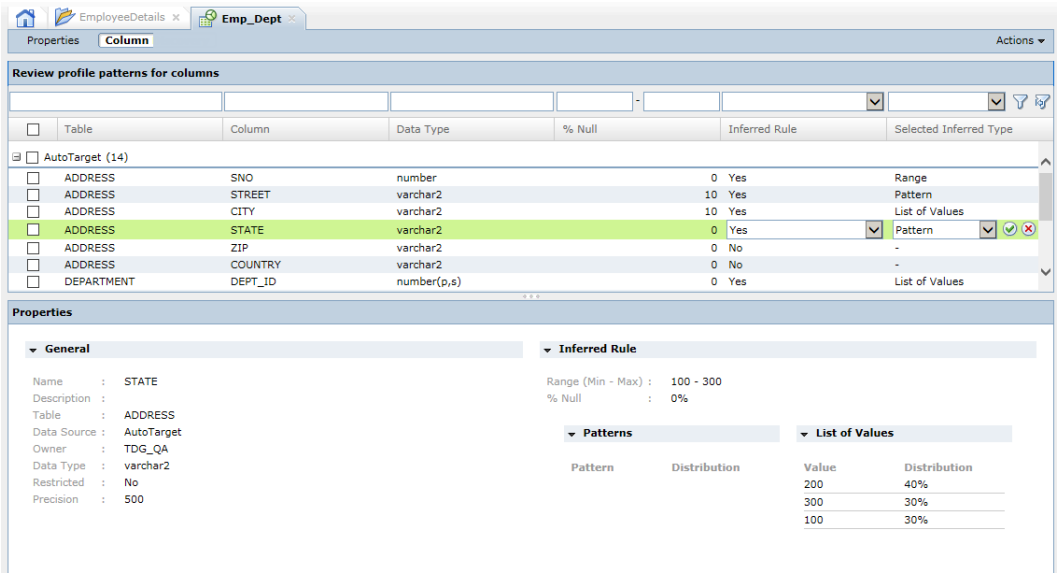
Column Name	Description
Table	Name of the table.
Source	Name of the column to mask.
% Data Conformance	The percentage of rows that contain metadata or data patterns that match the data domain.
% Null	The number of rows that contain NULL values in the source column.
Patterns	The number of data domain data or metadata patterns that match the column.
Column Name Match	Indicates if the column name is the matching data domain pattern.
Current Data Domain	Shows the previous data domain when you run the data domain profile more than once.
Profiled data domain	The data domain name returned from the latest profile run.
Status	Shows whether the data domain is verified or approved.

Column Profile Results

After you run a column profile, you can review the results and either accept or reject the results.

To view column profile results, click the **Profile | Column Discovery** view. Review inferred pattern for all the columns. Accept or reject profiled patterns for each column. Edit the profiled pattern for columns and reject unwanted patterns or specify a pattern. Select Yes in the Inferred Rule column, select the inferred type, and save the profiled pattern.

The following image shows the column profile results:



The following table describes the columns in the column profile results:

Column Name	Description
Table	Name of the table.
Column	Name of the target column to apply the profile result and generate test data.
Data Type	The data type of the target column.
% Null	The number of rows that contain NULL values in the source column.
Inferred Rule	Indicates that you can apply inferred rule types to the column. Default is a hyphen. Select Yes to apply the inferred rule.
Select Inferred Type	The type of inferred rule that is applicable for the column. You can select pattern, list of values, and range based on the data type of the column. Select a list of values or a data pattern for a string data type. Select a list of values or a range of values for a date data type. You can select a list of values, a data pattern, or a range of values for a numeric data type.

Project Tables

You can view the source tables in a project. You can view the primary keys and foreign keys in each table.

Click the **Discover | Tables** view to see the data source tables in the project. You can filter the list of tables by the table name or by the description.

Click the table name to view the table properties in the **General Properties** pane. View the columns in the table in the **Columns** view. Test Data Manager also shows the child tables in the **Relationship Overview** pane.

Table Classification

You can assign a classification to source tables to identify what kind of data the table contains. The classification does not affect profile operations. You can filter tables by this classification when you create entities or plans.

In the **Discover | Tables** view, select the tables that you want to assign a classification to. Choose one of the following table classifications:

- Temporary
- Master
- Configuration
- Transactional
- Log
- Seed

For example, you might want to know which tables are master tables when you determine which tables are driving tables for entities. You might want to configure more filter criteria for tables that contain transactions or logs.

Constraints

Foreign key constraints define parent-child relationships between the source tables. Use constraints to determine the tables to include in a data subset. You can also limit the values that you want to store in the data subset table columns.

When you create a data entity, you select one table for the entity. The table that you select is the entity driving table. Test Data Manager adds other tables to the entity based on the constraints for the driving table.

Use data discovery to find relationships between tables. When you identify the relationships that you want to add to the TDM repository, create and edit constraints in Test Data Manager.

The following types of constraints define relationships between tables in an entity:

Primary Key

A column or combination of columns that uniquely identifies a row in a data source. A table can have one primary key.

Logical Manual

A parent-child relationship between tables based on columns that are not keys. You can create the following logical constraints:

- Logical constraints that you accept from data discovery profiles. You can delete these logical constraints.
- Logical constraints that you define in Test Data Manager. You can delete these logical constraints.

Related Topics:

- "Entities" on page 108

Major and Minor Constraint Severity

When you define a foreign key constraint, configure the constraint severity as major or minor. The severity determines the scope of the data that a data subset receives based on the constraints. You can change the severity to change the final output of a subset operation. The constraint severity level does not affect the structure of an entity. You can change the severity level for a constraint after you create an entity. You can

also change the severity level at run time in a subset plan. Doing this does not change the severity levels in the entity.

Major constraints are constraints that move in both directions, child to parent and parent to child. When you want the parent table to select additional child records that are related to the parent, assign a major constraint between two tables.

When you define a constraint, you define it on the child table. When you define a constraint with a major severity level, all of the children of the parent are included in the data subset. For example, you define a constraint between Customer and Leads. Customer is the parent table. You create a filter on Leads to return only high-quality leads. The subset receives the customer for a high-quality lead, but it also receives all the leads for the customer. This major constraint honors referential and transactional integrity. You can apply only such a major constraint from within an entity.

The child record selected for the subset operation pulls in related parent records from both parent tables. The parent records select all the related child records from the child table. As more records are selected in the child table, the related parent records are selected again. This is because the records selected by one parent in the child table might refer to parent records in another parent that is not a part of the subset. This process continues in a loop until there are no more related records to select.

To configure a subset operation that aims to exclude additional data, you can select the Referential Integrity Only option in a plan at run time. When you select this option, the data included remains referentially intact, but this option does not guarantee that the subset will include all child records pertaining to a parent. Therefore, the subset operation might not maintain transactional integrity.

When you change the constraint definition at run time, the relations set in the entity do not change.

Minor constraints are unidirectional constraints, from child to parent. Minor constraints do not move in the reverse direction from parent to child. When you do not want the parent table to select additional child table records that are related to the parent, assign a minor constraint between two tables.

When you define a constraint with a minor severity level, the parent row is included in the data subset, but the data subset does not receive all of the children of the parent row. From the example above, the data subset receives the high-quality leads and the customers for the high-quality leads.

If the schema has constraints that form cycles, or if the schema has objects that have more than one major parent constraint, TDM performs recursive evaluation to calculate the subset. TDM needs a staging connection to perform recursive evaluation. You must therefore create a staging connection to perform subset operations on tables with cyclic or tending to cyclic constraints. It is recommended that you use the source connection as the staging connection.

Manually Add Keys to Tables in a Project

You can manually add primary keys and foreign keys to tables to establish relationships between tables for data subset operations. When you manually add keys, you do not need to run a primary key discovery profile.

When you add keys, you define constraints for data subset, data masking, and data generation operations in the project. You do not update the source database.

You can add the following types of keys constraints:

Primary Key

You can add one column or column combination as the primary key for a table. A primary key column cannot contain null or duplicate values. You cannot add more than one primary key constraint.

Foreign Key

Add a column or column combination as a foreign key in a table. When you define a constraint, you define a foreign key in a table and relate it to a column in a parent table.

Unique Key

Add a column or set of columns as a constraint to define a unique key in a table. A unique key column can contain null values. You can create more than one unique key in table. You can perform multiple data generation assignments when you filter the columns based on unique keys.

Conditional

Add a conditional constraint to perform a data generation operation. Create a conditional constraint to restrict the data that you want to generate in a column based on the data in another column. The conditional constraint that you enter becomes a part of the metadata of the target table. Configure expressions to apply constraints at the table level.

Creating a Primary Key Constraint

You can add a logical primary key to a table to create table relationships for data subset.

1. In a project, click the **Discover | Tables** view.
2. Click a table name to select the table.
3. Click **Constraints**.
4. Click **Create New Constraint**.

   The **New Constraint** dialog box appears.
5. Select **Primary Key**.
6. Click **Next**.
7. To add the columns, click **Add**. Select a column for which you want to add a primary key constraint.
8. Click **Finish**.

Creating a Logical Relationship Between Tables

You can add a logical foreign key to a table to create a table relationship for data subset. Choose a column from a parent table to establish a key relationship between the tables.

1. In a project, click **Discover** > **Tables**.
2. Click the table in which you want to create the foreign key.
3. Click the **Constraints** tab.
4. Click **Create New Constraint**.
5. Enter the constraint properties.

   The following table describes the constraint properties:

Property	Description
Name	Constraint identifier.
Constraint Type	Select the constraint type as Foreign Key. When you want to perform a data generation operation, you select the constraint type as Conditional.

Property	Description
Severity Level	Choose one of the following values: - Major. The relationship includes the parent table and it includes the other children of the parent. - Minor. The relationship includes the parent table but it does not include the other children of the parent.
Parent Table	Choose the parent table to establish the foreign key with.
Enable Constraint	Enable the foreign key relationship.

6. Click **Next**.

   A list of the columns in the table appears in the left panel. A list of the columns in the parent table appears in the right pane.
7. Click a child column from the left pane. Click a parent column from the right pane. Click the **Link** icon to map the parent-child relationship.
8. Click **Finish**.

Creating a Unique Key Constraint

You can add a unique key to a table to create table relationships for data subset and to perform bulk generation rule assignments at a time.

1. In a project, click the **Discover | Tables** view.
2. Click a table name to select the table.
3. Click **Constraints**.
4. Click **Create New Constraint**.

   The **New Constraint** dialog box appears.
5. Select **Unique Key**.
6. Click **Next**.
7. To add the columns, click **Add**. Select the column for which you want to add a unique key constraint.
8. Click **Finish**.

Creating a Conditional Constraint

Create a conditional constraint to restrict the data generated in a column based on the data in another column. Configure an expression to create a conditional constraint.

1. In a project, click **Discover** > **Tables**.
2. Click the table in which you want to create the conditional constraint.
3. Click the **Constraints** tab.
4. Click **Create New Constraint**.

   The **New Constraint** dialog box appears.
5. Enter a name for the conditional constraint.
6. Select the constraint type as Conditional.

7. Click **Next**.
8. Create an expression in the **Expression Builder**. Select the required columns, functions, variables, and operators to build expressions.
9. Click **Finish**.

CHAPTER 6

Creating a Data Subset

This chapter includes the following topics:

- Data Subset Overview, 106
- Data Subset Process Flow, 107
- Data Subset Components, 107
- Creating an Entity, 110
- Creating a Group, 115
- Applying Criteria to an Element or Attribute, 116
- Editing a Data Subset Component, 116
- Exporting a Data Subset Component, 117
- Importing a Data Subset Component, 117
- Copying a Data Subset Component, 118
- Deleting a Data Subset Component, 118
- Creating a Data Subset, 118
- Example - Data Subset for XSD Data Sources, 119

Data Subset Overview

You can create a subset of production data if you need a small, targeted, and referentially intact copy of the production data to use in a nonproduction environment. A nonproduction environment might include development, test, or training environments.

For example, you might create a subset of financial data for a specific region or time period.

You can create a data subset from relational, flat file, or XSD sources.

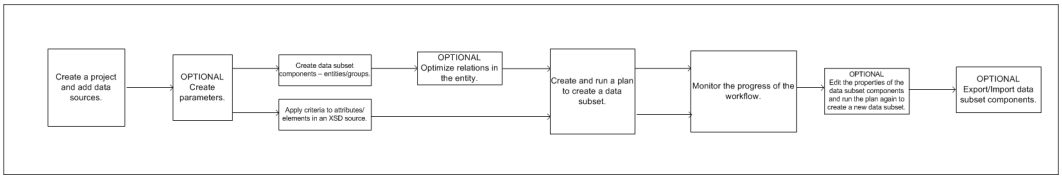
To create a data subset, you define data subset components and add them to a plan. The components that you create depend on the type of data source and the data subset that you require.

You can generate a workflow from the plan. The workflow loads a subset of the source data into a target database.

Data Subset Process Flow

Create a data subset to create a subset of production data that is referentially intact.

The following image shows the tasks that you perform to create a data subset and the tasks that you can perform on data subset components in Test Data Manager:



You can perform the following tasks related to data subset in Test Data Manager:

Create criteria parameters

You can create criteria parameters. Use criteria parameters in entity filter criteria. You can then run the same workflow to create data subsets based on multiple criteria values without updating the entity, plan, or creating another workflow.

Create data subset components

You can create entities and groups. Add filter criteria to an entity to create a data subset of related tables based on required criteria. Create a group to add one or more specific tables to a data subset.

Edit data subset components

You can edit entities and groups that you create or import and use the edited entities and groups to create a data subset. You can choose to optimize the relations between tables in an entity or choose to optimize relations when you run the workflow.

Export or import data subset components

You can export groups and entities that you create in Test Data Manager. You can import the entities and groups and use them to create a data subset with a different Test Data Manager Service.

Run a plan to create a data subset

You can add relational, flat file, or XSD sources to a plan to create a data subset based on criteria or subset components that you add to the plan.

Data Subset Components

You can create a data subset from relational, flat file, and XSD sources.

To create a data subset, you define one or more of the following components:

Entity

Defines a set of tables that are related based on physical or logical constraints. Create an entity when you need to copy data from related tables to a subset database. When you create an entity, select parameters to filter data in columns that you want to include in the subset database. You can manually create an entity or use an entity from profile results. Applicable for relational and flat file sources.

Group

Defines a set of unrelated tables. Create a group when you need to copy data from one or more unrelated tables to a subset database. Applicable for relational and flat file sources.

XSD Files

To create a data subset from an XSD source, you apply subset criteria to one or more elements or objects in the XSD file and define the scope of the criteria. Criteria determines the filter that you apply to an element or attribute. You can apply one filter for each hierarchy. The scope determines the level at which you apply the filter. The level at which you apply the criteria affects the outcome of the data subset operation.

After you apply the criteria and define the scope of the criteria, add the XSD file as the subset component to a plan.

Entities

An entity defines a structure for copying related data to the subset database.

An entity consists of a driving table and related tables. A driving table is the starting point for defining relationships between tables in the entity. Test Data Manager defines the tables that relate to the driving table based on physical and logical constraints. You can add tables that have cyclic or circular relationships within the table or between tables. You must add a staging connection before you can add these tables to an entity.

When you create an entity, you select a table as the driving table. Test Data Manager retrieves all related tables to the driving table based on the constraints in the driving table. Test Data Manager adds the related tables to the entity by default.

Define filter parameters for columns of tables in the entity. The filter parameters define the criteria for the subset of the data. To define the subset criteria, you can enter a specific value or select a parameter from a list of criteria parameters that you create in the project. You can change the parameter value when you run the workflow to create test data.

You can select a relationship in an entity and specify the minimum and maximum number of child records to generate test data.

Related Topics:

- "Constraints" on page 101

Entity Views

Create, edit, and delete entities in the **Define | Data Subset** view of a project.

When you create an entity, you define a driving table for the entity. The driving table is a starting point for determining the structure of the entity. Test Data Manager determines the rest of the tables in the entity based on the constraints that include the driving table.

When you create an entity, you can configure it with the following views:

Entity Map

The **Entity Map** view is a graphical representation of the tables in the entity and the relationships between the tables. You can disable and enable relationships, and edit filter criteria for each table in the **Entity Map** view.

If you drag the entity map panel to the left, a **Table Properties** panel appears that includes information about a selected table or constraint. If you click the link between two tables, the **Constraint Properties** panel appears.

You can change which tables you view in the entity map. You can view all the tables or you can view the tables by their relationships with a selected table. By default the entity map shows enabled relationships.

You can choose the **Show excluded tables** option with any of the views to show disabled relationships.

Select one of the following views:

All

Shows all tables in the entity.

All parents

Shows all the parent tables of a selected table.

All immediate relationships

Shows the tables that have constraint relationships with a selected table. You can view one level of parent and child tables from the selected table.

All children

Shows all the child tables of a selected table.

All cyclic relationships

Shows all constraint relationships that create cyclic relationships. The maximum number of cyclic relationships that TDM displays is 1000. You can choose to hide nodes and paths that are not a part of the cycles.

You can select any table in the map to edit the filter criteria. You can click a link to edit the constraint between the tables. You can disable relationships between tables from the map. When you select links in the **Entity Map** view, the corresponding rows are selected in the **Relationships** view.

You might define the driving table as Employee. Employee has a primary key, EmpID. Employee has three child tables that have the EmpID foreign key. The EmployeeBankDetails child table has an AccountNum constraint that relates EmployeeBankDetails to a BankAccounts table. When you create an entity with Employee as the driving table, Test Data Manager might create an entity map similar to the following map:

The driving table icon contains a star in the entity map.

To add a filter criteria for Employee, click the Employee table in the entity map. Edit the filter criteria from the Table Properties panel.

When you click the links between the Employee, EmployeeBankDetails, and BankAccounts table, the corresponding rows are selected in the **Relationships** view.

Tables

The **Tables** view contains a list of the tables in the entity. You can edit filter criteria for each table in the **Tables** view.

The **Tables** view shows the table name, description, and the name of the database that contains the table. The status field indicates whether the entity includes or excludes the table. When you disable a relationship in the entity and the table is not related to any table in the entity map, the table is excluded from the data subset database.

Relationships

The **Relationships** view shows a list of relationships in the entity. For each constraint, the view shows the parent table, the child table, and the database names for each table in the relationship. You can disable and enable relationships between tables in the **Relationships** view.

When a table has more than one relationship in an entity, the table appears more than once in the view. The **Status** field indicates whether the relationship is enabled or disabled.

When you select a row in the **Relationships** view, the corresponding link is selected in the **Entity Map** view. To generate test data, you can select a relationship and specify the number of child records that you want in the target.

Groups

A group defines one or more unrelated tables that you want to copy to a subset database. Create a group to add unrelated tables to a plan, or to copy unfiltered data to a target.

When you add tables to a group, you can include residual tables in the group. A residual table is a table that has not been added to a group or entity. Select residual tables when you want to include all the tables in a source in a data subset plan.

When you edit a group, you can add or remove tables from the group.

You create, edit, and delete groups in the **Define** view of an application.

Group Example

The test data contains multiple tables with organization information and employee information.

Some of the tables contain foreign keys and are related tables. Some of the tables that contain employee information are not related to tables that contain organization information. You want to create a data subset that contains employee skill information and organization location information. To ensure that you include all tables with data that you require in the data subset, create a group and add the tables that contain the data.

Create and run a plan that includes the group. The data subset that you create contains tables with the data that you require.

Creating an Entity

When you create an entity, you select the driving table. Test Data Manager retrieves the related tables based on the constraints. You can add filter criteria to the driving table to filter source data when you create the data subset database. You can disable relationships between tables in the entity.

To use parameters in an entity, you must create the entities in the project before you create the entity. Before you create an entity, perform data discovery to identify constraints. Add constraints to the source to define the child tables.

1. Open a project.
2. Click **Define** > **Data Subset**.
3. Click **Actions** > **New** > **Entities**.
4. In the **New Entity** dialog box, enter a name and optional description for the entity.
5. Click **Select Table** to select a driving table from the list and click **OK**.
6. Click **Save** to create the entity.

   Test Data Manager displays a diagram of the relationship between the tables in the entity. You can view a list of the tables or a list that shows the relationships between the tables.
7. To create a subset of the data based on filter criteria, click **Edit** in the **Properties** pane.

The **Edit Criteria** dialog box appears.

8. To add criteria for a column, click the **Entity Criteria** tab.
9. Click the **Add Criteria** button.
10. Select the columns on which you want to filter data and click **OK**.

    The columns that you selected appear in the columns list in the **Entity Criteria** tab.
11. To define the filter expression in the entity, select the attribute from the Attribute list. Choose **Value** to enter a value or **Parameter** to enter a parameter name.

    You can indicate a filter condition in the entity, but define the expression in the plan. You can choose to create multiple criteria with parameters and include required parameters when you run a workflow.
12. Select an operator from the list to filter the data.
13. If you enter a filter condition, enter a specific value or choose the variable from the list of variables to complete the filter expression.
14. Click **Save** for each filter criteria that you create. If you define multiple filters in an entity, the filter conditions act as "AND."
15. Click **Save** to exit the **Edit Criteria** dialog box.

Entity Constraint Properties

You can configure the constraint properties between two tables in an entity map.

The following table describes the constraint properties that you can configure in an entity map:

Property	Description
Name	The name of the relationship. You cannot edit the name.
Type	The type of the relationship. Select Major or Minor relationship type.
Input Type	The physical or logical type of input. You cannot edit the input type.
Parent Table	The driving table or the parent table. You cannot edit the parent table.
Child Table	The related table or the child table. You cannot edit the child table.
Minimum	The minimum number of child records that you want to generate for each parent record.
Maximum	The maximum number of child records that you want to generate for each parent record.
Status	The status of the relationship. You can enable or disable the relationship.

Parameters in Entity Criteria

You can use parameters to specify the filter criteria to create a data subset. Parameters allow you to change values each time you run a workflow without editing a plan. You can create criteria parameters in a project and maintain the parameter values in the project or in an external parameter file. A parameter file is a file that lists user-defined parameters and their values.

When you use parameters to specify the filter criteria in an entity, you select a parameter name instead of entering a specific value in the criteria.

For example, you want to create a data subset that includes all rows with the EMPID greater than 100. Instead of entering the value as 100 in the value field, you select the name of the parameter. Enter the parameter value as 100 in the parameter file.

Create a criteria parameter $$CRITERIAVALUE in the project. If you include a parameter file in the workflow that you run, the parameter file must contain an entry $$CRITERIAVALUE=100. Enter the criteria EMPID > $ $CRITERIAVALUE in the entity.

To create a data subset with a different filter criteria value, you can change the value of the parameter and run the same workflow. To create a data subset that contains rows with EMPID greater than 500, change the parameter value to 500 and then run the workflow.

Enter the parameter file name and the file path when you configure the plan and run the workflow.

When you use parameters for a string field or a date field on a relational source, you must enclose the value in single quotes in the parameter file.

You can include multiple filter criteria parameters in an entity. You can choose to include the criteria parameters that you want each time you run the workflow. If you maintain the parameter values in the project, choose to include the criteria parameters that you want in the **Execute Workflow** window.

If you maintain the parameter values in a parameter file that you create, you must create ignore flags for each criteria parameter. Set the flag to "Y" or "N" to ignore the criteria or consider the criteria in a workflow. A workflow does not consider criteria that you flag for ignore.

For example, the criteria parameter $$CRITERIAVALUE must have a corresponding $ $CRITERIAVALUE_IGNORE flag. Set the flag value to "N" to include the criteria in a workflow. Set the flag to "Y" to ignore the criteria in the workflow.

If you create the parameter file in Test Data Manager, the file contains ignore flags for each criteria parameter. By default, the flag is set to "N".

Database Functions in Entity Criteria

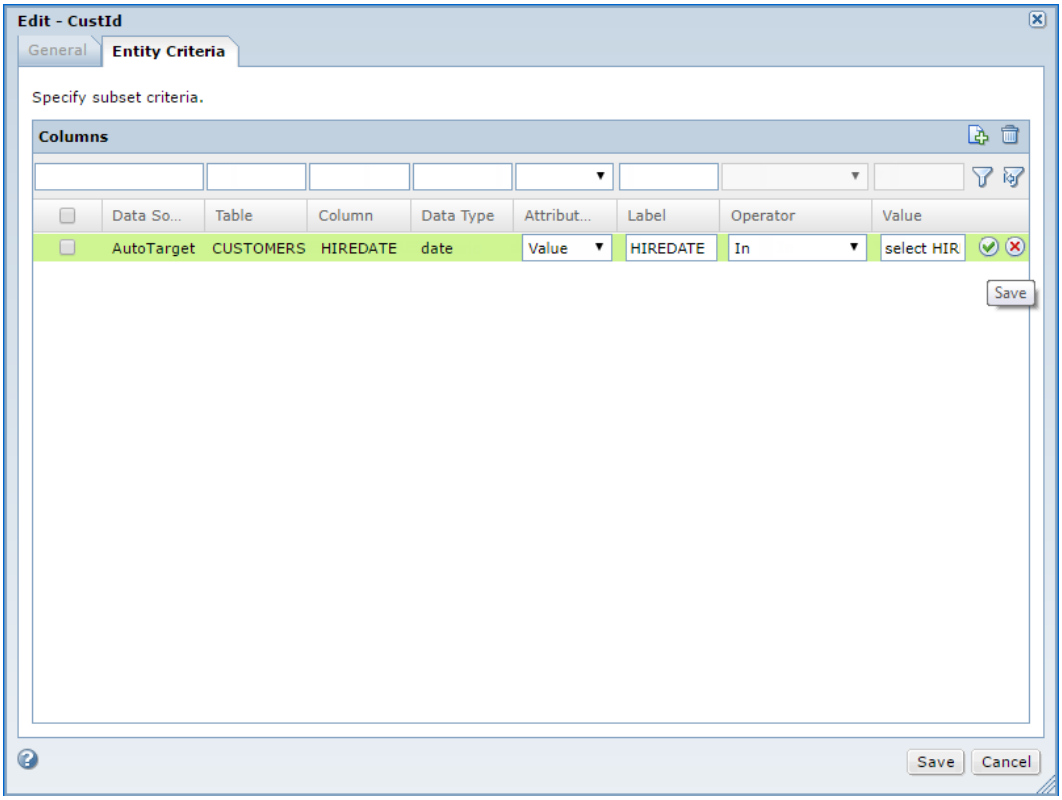
You can use database functions in entity filter criteria. To ensure that TDM processes the filter query correctly, you must use the IN operator and a Select clause.

For example, consider filter criteria that uses the add_months database function. Workflow generation fails if the plan contains the query entered directly, as TDM does not recognize this as a database function. TDM is unable to pass the criteria with the database function as is to the database.

To use the filter criteria with a database function, use the IN operator and enter the query with a Select clause.

To add the required filter criteria, in the **Edit Criteria** dialog box, select the operator IN from the list of operators. Enter the database function as a Select clause in the **Value** field.

```
select HIREDATE from <table> where HIREDATE > add_months(current_date, -3)
```



TDM processes the filter criteria correctly and passes the query to the database. The workflow runs successfully with this entity criteria.

Optimizing Relations

By default, all constraint relationships within an entity are major. This can sometimes result in including excess data in the subset, when you run a subset operation. Optimize the relations such that the entity has a minimum number of major relations to ensure the subset operation fetches minimal data while considering all tables in the entity. You can also optimize relations at run time in a plan. When you optimize relations at run time, you do not save the changes to the entity.

The entity must have a criteria before you can optimize relations in the entity. TDM generates an error if you optimize relations without assigning criteria at the entity level.

Perform the following steps to optimize relations in an entity:

1. Open the entity.
2. Click **Actions** > **Optimize Relations**.

   TDM analyzes the constraints in the entity and proposes major relationships or minor relationships based on analysis. The **Optimize Relationships** page displays the proposed relationship between each parent table and child table along with the current relationship.
3. Optional. Click **Download Optimizations** to download and save the optimization information in a CSV file.
4. Click **Optimize** to accept all the proposed relationships. You cannot apply results selectively .

A list of any plans that the entity is included in appears. To update the changes in the plans, generate and run the plans again.

5. Optional. To download the list of plans in a .csv file, click **Export**.

   The file contains a list of the plans with the description and the project information.

6. Click **Continue**.

   The entity is updated with the optimized relations. You can manually edit any required constraints even after you save the changes.

Disabling and Enabling Relationships in an Entity

You can enable or disable relationships in an entity from the Entity Map view or the Relationships view.

1. In Test Data Manager, open a project.
2. Navigate to the **Define | Data Subset** view.
3. Click the entity to open the **Entity Map** view.
4. You can edit a relationship in the **Entity Map** view or the **Relationships** view.
   - To edit a relationship in the **Entity Map** view:
     1. Choose to view all relationships and to show excluded tables. An enabled relationship appears as a solid line and a disabled relationship appears as a dotted line.
     2. Click the link between the parent and child icons in the map to select the link. Test Data Manager highlights the link.
     3. Click **Disable Relationship** to disable the relationship or **Enable Relationship** to enable the relationship.
   - To edit a relationship in the Relationships view:
     1. Click the **Relationships** view.
     2. Click **Disable Relationship** or **Enable Relationship**. The status changes to disabled or enabled.

Review Changes

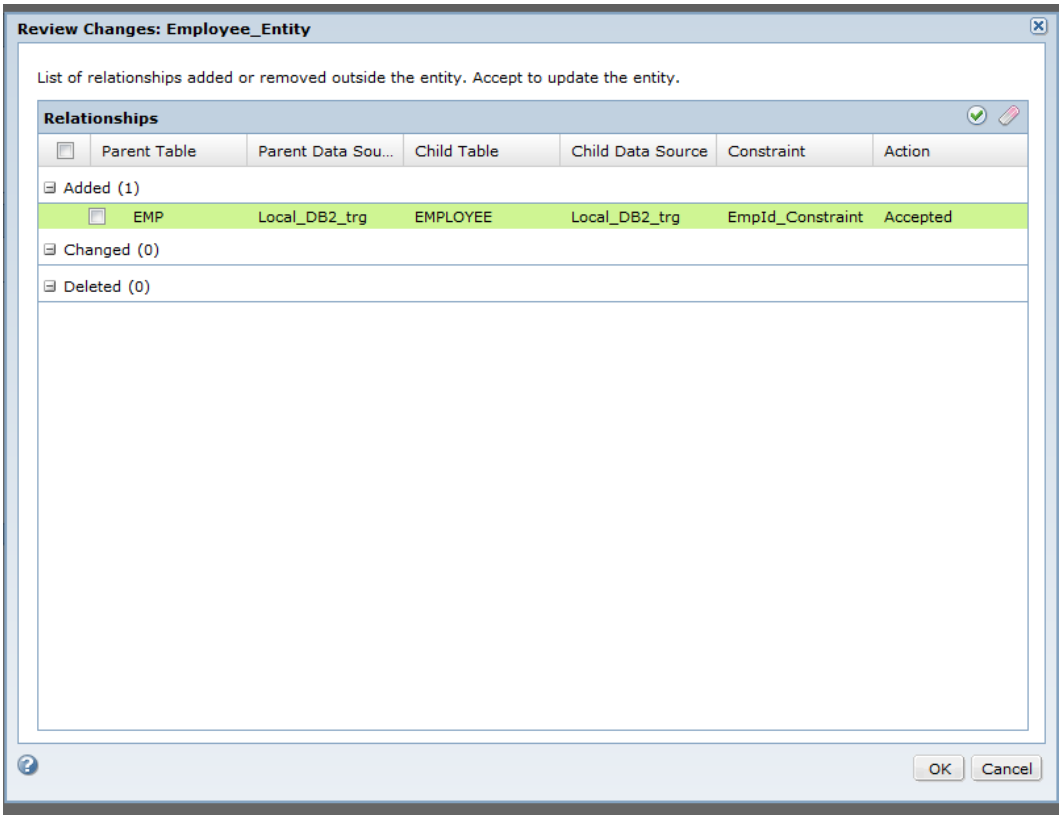
When you modify a constraint that affects a table in an entity, you must update the entity to reflect the change.

For example, you create an entity that contains a table called EMP. After you create the entity, you add a constraint relationship between EMP and another table. When you add the constraint, the additional table does not appear in the entity unless you update the entity relationships.

When you view the list of entities in a project, Test Data Manager flags the entity as having a change impact. Open the entity and click **Review Changes** to see the relationship changes that impacted the entity. You can select a relationship and accept the change to update the entity.

Note: After you review and update the entity, you must indicate that you are finished reviewing the changes. In the Entity view, click **Actions** > **Mark Changes Review as Complete**. The Review Changes notification no longer appears for the entity.

The following image shows the **Review Changes** dialog box:



Entity Example

An employee has an ID number of 10 and you want information about EmpID 10 from multiple tables. A table called Employee contains an EmpID column. The Employee_History and Employee_Order tables have constraints that relate them to the Employee table that contain the EmpID column.

Create an entity and define Employee as the driving table. Test Data Manager adds the Employee_History and Employee_Order tables to the entity. Define a criteria in which the EmpID value must be equal to 10.

Create a plan, generate the workflow, and start the workflow. The target database contains all the data for EmpID 10. The Employee table has one row with EmpID 10, the Employee_History table has multiple rows of history for EmpID 10, and Employee_Order has all the orders for EmpID 10.

Creating a Group

To create a group, select the tables that you want to add to a data subset plan.

1. On the **Define | Data Subset** view in the project, click **Actions** > **New** > **Groups**.
2. In the **Create Group** dialog box, enter a name for the group and optional description for the group. Click **Next**.
3. To select one or more tables for the group, click **Add Tables**.
4. Optionally, you can filter the list of tables to search for.

5. Select the required tables and click **OK**.
6. Click **Save**.

Applying Criteria to an Element or Attribute

Apply subset criteria and specify the scope of the criteria to create a data subset from an XSD source. The scope and criteria determine the output of a data subset operation. You can apply criteria to elements and attributes in an XSD file.

Import the XML source definition from the PowerCenter repository into the project before you apply the criteria.

1. Open a project and click **Discover** > **Files**.

   A list of XSD files in the project appears.
2. To open the required file in a separate page, select the file.
3. Select the element or attribute on which you want to apply the subset criteria.
4. Click **Edit** on the **Criteria** tab of the **Details** tab.

   The **Edit Criteria** window opens.
5. Select the required operator from the list of operators.
6. Enter the value or set of values for the criteria.

   You cannot enter a parameter.
7. Select the scope from the list of options and click **OK**.

   The criteria appears in the **Criteria** tab. To clear criteria, select **Criteria** from the list and click the **Clear** button.

Add the XSD file with the scope and criteria assignment to a plan. Run the plan and workflow to create a data subset with the required data.

Editing a Data Subset Component

You can edit an entity or group in a project. Edit an entity to change the general properties, relationships, and filter criteria for tables. Edit a group to change the general properties and tables in the group.

1. In a project, click **Define | Data Subset**.

   The **Data Subset** tab shows a list of the entities and the groups in the project.
2. Click the required entity name or group name to open it.
3. Click **Actions** > **Edit**.

   The **Edit** dialog box appears.
4. On the **General** tab, edit the name, description, and status.
5. To edit an entity, on the **Entity Criteria** tab, select columns and enter filter criteria.
6. To edit a group, on the **Tables** tab, change the tables in the group.
7. Click **Save**.

A list of plans that include the entity or group appears. To update the changes in the plans, generate and run the plans again.

8. Optional. To download the list of plans in a .csv file, click **Export**.

   The file contains a list of the plans with the description and the project information.

9. Click **Continue**.

Exporting a Data Subset Component

You can export an entity or group to an XML file and import the XML file to another TDM repository.

1. Click **Projects** to open the **Projects** view.

   A list of projects appears.

2. Open the project that contains the entity or group to export.
3. Click the **Define** view.

   The list of entities and groups in the project appears.

4. Select an entity or group to export.

   Use the checkbox to select the required component.

5. Click **Actions** > **Export**.
6. Choose the name and path of the XML file to create.

   The default name is a string that contains "<component type>_" and the current date and the time.

7. Click **OK**.

Importing a Data Subset Component

You can import an entity or group from an XML file that was exported from another TDM repository.

1. To open the **Projects** view, click **Projects**.
2. Click the project description to select a project to import the entity into.

   Do not open the project.

3. Click **Actions** > **Import**.
4. Browse for the XML file that contains the entity or group to import.

   The XML file has a default name with a string that contains "<component type>_" and the date and time of export.

5. To import the entity or group, click **Finish**.

Copying a Data Subset Component

You can create a data subset component by copying a data subset component. Test Data Manager copies the tables in the original group or entity into the new group or entity.

1. Open a project and click **Define** > **Data Subset** to view a list of subset components in the project.
2. Click the description to select the required subset component.

   Do not open the entity or group.
3. Click **Actions** > **Duplicate**.

   The **Duplicate <component name>** dialog box appears.
4. Change the name and description. Click **Save**.

Deleting a Data Subset Component

If you do not use a subset component, you can delete the component. If you delete a subset component that is assigned to a plan, the plan is not valid.

1. In the **Define** | **Data Subset** view in the project, click to select the required data subset components.

   Do not open the subset component.
2. Click the **Actions** > **Delete**.
3. In the **Delete Objects** message box, click **OK**.

   A list of plans that the subset components are included in appears. To update the changes in the plans, generate and run the plans again.
4. Optional. To download the list of plans in a .csv file, click **Export**.

   The file contains a list of the plans with the description and the project information.
5. Click **Continue**.

Creating a Data Subset

Create a data subset to create a referentially intact subset of production data.

1. Create a project and add the required data sources to the project.

   Create different projects to work with XSD sources and relational sources.
2. Optional. Create connection, owner, or criteria parameters to use in the plan and entity.
3. Perform the following tasks based on the type of data source:
   - Relational database. Create the required data subset components for relational sources.
   - Flat files. Create the required data subset components for relational sources.
   - XSD files. Assign the required criteria to elements and attributes in the XSD file.
4. Optional. Optimize the relationships in the entity.
5. Create a plan and add the required data subset components and data sources to the plan.

6. Run the workflow.
7. Monitor the progress of the workflow.

Related Topics:

- "Creating a Project" on page 52
- "Importing Data Sources" on page 60
- "Plan Settings" on page 268
- "Creating a Data Masking and Data Subset Plan" on page 290
- "Workflow Generation" on page 293
- "Executing a Workflow" on page 295
- "Viewing the Log Messages" on page 304

Example - Data Subset for XSD Data Sources

A product test team maintains its test data in a master XML file. To test specific features of the product, different test groups require specific test data. Test group A requires data on a specific skill set Java. Test group B requires data on a specific location BLR.

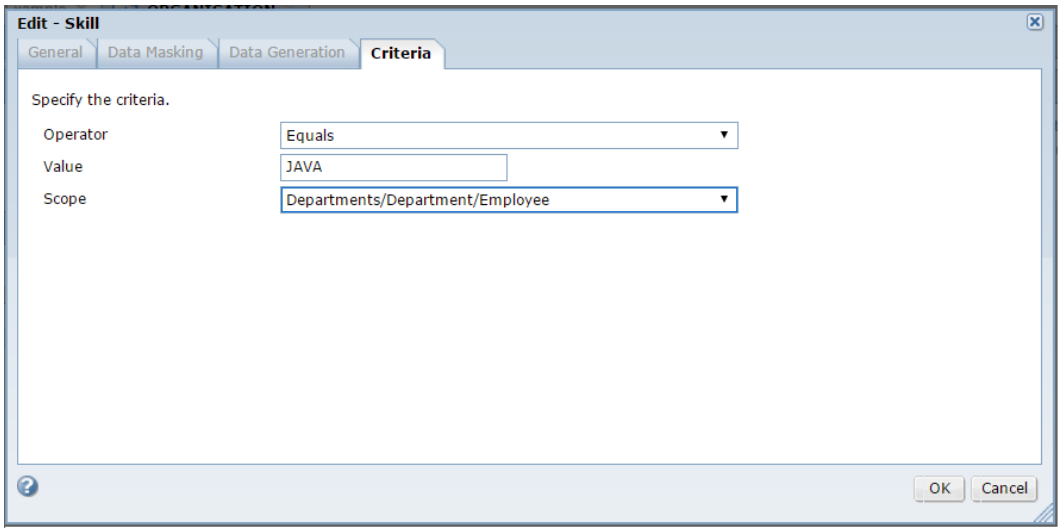
To create different test data files for each test group, the test team uses TDM to create data subsets with the required information.

The master data file contains records of employee and contractor information including name, department, address, office location, and skill set.

The XML source definition file (XSD) defines the structure of the master file and data subset files that you create from the master data file. The XSD file defines the Street, Zip, City, and State elements as child elements of the element Address. The elements Name, Skill, and Address are child elements of the elements Employee and Contractor. Address, Employee, and Contractor are child elements of the element Department.

Ancestor and child relations affect the outcome of a data subset operation.

The following image shows the **Criteria** tab on the file page in Test Data Manager:



To create a data subset that contains employee information for the skill set Java, apply the criteria to the element Skill on the Employee branch. Select the operator as equals. Enter the value on which to filter the data as JAVA. Define the scope as Employee.

Create and run a plan that includes the XSD file and the criteria and scope of the data subset.

The subset operation applies the filter on the element Employee and includes all employees with skills that match the criterion. The data subset file that the workflow creates contains data on all employees with Java skills. Department is an ancestor element of Employee. The subset therefore includes all departments that have employees with Java skills and all child elements of Departments.

Edit the data subset criteria and then run the plan again to create a data subset with the data that test group B requires. Delete the criteria on the element Skill.

To create a data subset of departments located in BLR, apply the filter criteria to the element City. Select the operator as equals. Enter the value on which to filter the data as BLR. Define the scope as Department.

The subset operation applies the filter on the element City and includes all departments with addresses that have cities that match the criterion. The data subset file that the workflow creates contains data on all departments with the city as BLR. The subset includes all child elements of the element Department.

Example XML Structure Definition

Consider the following XSD structure of the master test data file:

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="Skill" type="xsd:string"/>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="Street" type="xsd:string"/>
    <xsd:element name="State" type="xsd:string"/>
    <xsd:element name="City" type="xsd:string"/>
    <xsd:element name="Zip" type="xsd:string"/>
    <xsd:attribute name="id" type="xsd:integer"/>
    <xsd:attribute name="name" type="xsd:string"/>
    <xsd:attribute name="location" type="xsd:string"/>

    <xsd:element name="Address">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Street" minOccurs="1" maxOccurs="1"/>
                <xsd:element ref="Zip" minOccurs="1" maxOccurs="1"/>
                <xsd:element ref="City" minOccurs="1" maxOccurs="1"/>
                <xsd:element ref="State" minOccurs="1" maxOccurs="1"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="Employee">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Name" minOccurs="1" maxOccurs="1"/>
                <xsd:element ref="Address" minOccurs="1" maxOccurs="1"/>
                <xsd:element ref="Skill" minOccurs="1" maxOccurs="unbounded"/>
            </xsd:sequence>
        <xsd:attribute ref="id"/>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="Department">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Address" minOccurs="1" maxOccurs="1"/>
                <xsd:element ref="Employee" minOccurs="1" maxOccurs="unbounded"/>
                <xsd:element ref="Contractor" minOccurs="1" maxOccurs="unbounded"/>
            </xsd:sequence>
```

```
            <xsd:attribute ref="name"/>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="Departments">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Department" minOccurs="1" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute ref="location"/>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="Contractor">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Name" minOccurs="1" maxOccurs="1"/>
                <xsd:element ref="Address" minOccurs="1" maxOccurs="1"/>
                <xsd:element ref="Skill" minOccurs="1" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute ref="id"/>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Example Master XML Data

Consider the following master test data file:

```
?xml version="1.0" encoding="US-ASCII" standalone="no"?>
<Departments location="INDIA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Department name="QA">
    <Address>
      <Street>VBIT</Street>
      <Zip>500081</Zip>
      <City>HYD</City>
      <State>TELANGANA</State>
    </Address>
    <Employee id="101">
      <Name>RAMA</Name>
      <Address>
        <Street>VBIT</Street>
      <Zip>500081</Zip>
      <City>HYD</City>
      <State>TELANGANA</State>
      </Address>
      <Skill>SELENIUM</Skill>
    </Employee>
    <Employee id="102">
      <Name>KRISHNA</Name>
      <Address>
        <Street>VBIT</Street>
      <Zip>500081</Zip>
      <City>HYD</City>
      <State>TELANGANA</State>
      </Address>
      <Skill>JAVA</Skill>
    </Employee>
    <Contractor id="901">
      <Name>RAMESH</Name>
      <Address>
        <Street>VBIT</Street>
      <Zip>500081</Zip>
      <City>HYD</City>
      <State>TELANGANA</State>
      </Address>
      <Skill>TESTING</Skill>
    </Contractor>
    <Contractor id="902">
```

```
    <Name>PAVAN</Name>
    <Address>
      <Street>VBIT</Street>
    <Zip>500081</Zip>
    <City>HYD</City>
    <State>TELANGANA</State>
    </Address>
    <Skill>JAVA</Skill>
  </Contractor>
</Department>
<Department name="DEV">
  <Address>
    <Street>INFA</Street>
    <Zip>500030</Zip>
    <City>BLR</City>
    <State>KARNATAKA</State>
  </Address>
  <Employee id="201">
    <Name>ANAY</Name>
    <Address>
      <Street>INFA</Street>
    <Zip>500030</Zip>
    <City>BLR</City>
    <State>KARNATAKA</State>
    </Address>
    <Skill>MAPGEN</Skill>
  </Employee>
  <Contractor id="903">
    <Name>AJEET</Name>
    <Address>
      <Street>INFA</Street>
    <Zip>500030</Zip>
    <City>BLR</City>
    <State>KARNATAKA</State>
    </Address>
    <Skill>JAVA</Skill>
  </Contractor>
</Department>
<Department name="DEVOPS">
  <Address>
     <Street>VBIT</Street>
    <Zip>500081</Zip>
    <City>HYD</City>
    <State>TELANGANA</State>
  </Address>
  <Employee id="301">
    <Name>NARAYAN</Name>
    <Address>
     <Street>VBIT</Street>
    <Zip>500081</Zip>
    <City>HYD</City>
    <State>TELANGANA</State>
    </Address>
    <Skill>BUILD</Skill>
  </Employee>
  <Contractor id="904">
    <Name>SHILPA</Name>
    <Address>
       <Street>VBIT</Street>
    <Zip>500081</Zip>
    <City>HYD</City>
    <State>TELANGANA</State>
    </Address>
    <Skill>BUILD</Skill>
  </Contractor>
  <Contractor id="905">
    <Name>AJEET</Name>
    <Address>
      <Street>INFA</Street>
    <Zip>500030</Zip>
    <City>BLR</City>
```

```
    <State>KARNATAKA</State>
    </Address>
    <Skill>JAVA</Skill>
  </Contractor>
</Department>
<Department name="DBA">
  <Address>
    <Street>INFA</Street>
    <Zip>500030</Zip>
    <City>BLR</City>
    <State>KARNATAKA</State>
  </Address>
  <Employee id="402">
    <Name>REKHA</Name>
    <Address>
      <Street>INFA</Street>
    <Zip>500030</Zip>
    <City>BLR</City>
    <State>KARNATAKA</State>
    </Address>
    <Skill>SQL_SYBASE</Skill>
  </Employee>
  <Employee id="402">
    <Name>RAMESH</Name>
    <Address>
      <Street>VBIT</Street>
    <Zip>500081</Zip>
    <City>HYD</City>
    <State>TELANGANA</State>
    </Address>
    <Skill>ORACLE</Skill>
  </Employee>
  <Contractor id="907">
    <Name>RAMANA</Name>
    <Address>
      <Street>INFA</Street>
    <Zip>500030</Zip>
    <City>BLR</City>
    <State>KARNATAKA</State>
    </Address>
    <Skill>JAVA</Skill>
  </Contractor>
  <Contractor id="906">
    <Name>VASU</Name>
    <Address>
      <Street>INFA</Street>
    <Zip>500030</Zip>
    <City>BLR</City>
    <State>KARNATAKA</State>
    </Address>
    <Skill>TERADATA</Skill>
  </Contractor>
</Department>
<Department name="HR">
  <Address>
     <Street>INFA</Street>
    <Zip>500030</Zip>
    <City>BLR</City>
    <State>KARNATAKA</State>
  </Address>
  <Employee id="501">
    <Name>SURENDRA</Name>
    <Address>
       <Street>INFA</Street>
    <Zip>500030</Zip>
    <City>BLR</City>
    <State>KARNATAKA</State>
    </Address>
    <Skill>RECRUIT</Skill>
  </Employee>
  <Contractor id="907">
```

```
      <Name>LAVANYA</Name>
      <Address>
         <Street>VBIT</Street>
      <Zip>500081</Zip>
      <City>HYD</City>
      <State>TELANGANA</State>
      </Address>
      <Skill>RESOURCE</Skill>
    </Contractor>
  </Department>
</Departments>
```

Data Subset for Group A

Run the plan the first time with the criteria to create a data subset that meets the requirements of Group A.

The output of the subset operation includes the following data:

```
<?xml version="1.0" encoding="US-ASCII" standalone="no"?>
<Departments location="INDIA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Department name="QA">
    <Address>
      <Street>VBIT</Street>
      <Zip>500081</Zip>
      <City>HYD</City>
      <State>TELANGANA</State>
    </Address>

    <Employee id="102">
      <Name>KRISHNA</Name>
      <Address>
        <Street>VBIT</Street>
        <Zip>500081</Zip>
        <City>HYD</City>
        <State>TELANGANA</State>
      </Address>

      <Skill>JAVA</Skill>
    </Employee>

    <Contractor id="901">
      <Name>RAMESH</Name>
      <Address>
        <Street>VBIT</Street>
        <Zip>500081</Zip>
        <City>HYD</City>
        <State>TELANGANA</State>
      </Address>

      <Skill>TESTING</Skill>
    </Contractor>
    <Contractor id="902">
      <Name>PAVAN</Name>
      <Address>
        <Street>VBIT</Street>
        <Zip>500081</Zip>
        <City>HYD</City>
        <State>TELANGANA</State>
      </Address>

      <Skill>JAVA</Skill>
    </Contractor>
  </Department>
</Departments>
```

Data Subset for Group B

Run the plan again with the criteria to create a data subset that meets the requirements of Group B.

The output of the subset operation includes the following data:


```
<?xml version="1.0" encoding="US-ASCII" standalone="no"?>
<Departments location="INDIA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Department name="DEV">
    <Address>
      <Street>INFA</Street>
      <Zip>500030</Zip>
      <City>BLR</City>
      <State>KARNATAKA</State>
    </Address>

    <Employee id="201">
      <Name>ANAY</Name>
      <Address>
        <Street>INFA</Street>
        <Zip>500030</Zip>
        <City>BLR</City>
        <State>KARNATAKA</State>
      </Address>

      <Skill>MAPGEN</Skill>
    </Employee>

    <Contractor id="903">
      <Name>AJEET</Name>
      <Address>
        <Street>INFA</Street>
        <Zip>500030</Zip>
        <City>BLR</City>
        <State>KARNATAKA</State>
      </Address>

      <Skill>JAVA</Skill>
    </Contractor>
  </Department>
  <Department name="DBA">
    <Address>
      <Street>INFA</Street>
      <Zip>500030</Zip>
      <City>BLR</City>
      <State>KARNATAKA</State>
    </Address>

    <Employee id="402">
      <Name>REKHA</Name>
      <Address>
        <Street>INFA</Street>
        <Zip>500030</Zip>
        <City>BLR</City>
        <State>KARNATAKA</State>
      </Address>

      <Skill>SQL_SYBASE</Skill>
    </Employee>
    <Employee id="402">
      <Name>RAMESH</Name>
      <Address>
        <Street>VBIT</Street>
        <Zip>500081</Zip>
        <City>HYD</City>
        <State>TELANGANA</State>
      </Address>

      <Skill>ORACLE</Skill>
    </Employee>
```

```
    <Contractor id="907">
      <Name>RAMANA</Name>
      <Address>
        <Street>INFA</Street>
        <Zip>500030</Zip>
        <City>BLR</City>
        <State>KARNATAKA</State>
      </Address>

      <Skill>JAVA</Skill>
    </Contractor>
    <Contractor id="906">
      <Name>VASU</Name>
      <Address>
        <Street>INFA</Street>
        <Zip>500030</Zip>
        <City>BLR</City>
        <State>KARNATAKA</State>
      </Address>

      <Skill>TERADATA</Skill>
    </Contractor>
  </Department>
  <Department name="HR">
    <Address>
      <Street>INFA</Street>
      <Zip>500030</Zip>
      <City>BLR</City>
      <State>KARNATAKA</State>
    </Address>

    <Employee id="501">
      <Name>SURENDRA</Name>
      <Address>
        <Street>INFA</Street>
        <Zip>500030</Zip>
        <City>BLR</City>
        <State>KARNATAKA</State>
      </Address>

      <Skill>RECRUIT</Skill>
    </Employee>

    <Contractor id="907">
      <Name>LAVANYA</Name>
      <Address>
        <Street>VBIT</Street>
        <Zip>500081</Zip>
        <City>HYD</City>
        <State>TELANGANA</State>
      </Address>

      <Skill>RESOURCE</Skill>
    </Contractor>
  </Department>
</Departments>
```

CHAPTER 7

Performing a Data Masking Operation

This chapter includes the following topics:

- Data Masking Overview, 127
- Data Masking Task Flow, 128
- Data Masking Rules, 128
- Creating and Assigning Data Masking Rules, 131
- Modifying Data Masking Rules and Assignments, 136
- Performing a Data Masking Operation, 140
- Data Masking Components, 141

Data Masking Overview

Use data masking to replace source data in sensitive columns with realistic test data for nonproduction environments. When you create data masking rules, you define the logic to replace sensitive data. To configure the sensitive columns that you want to mask, assign data masking rules to source columns, data domains, and policies.

A policy defines the data masking rules, the data to mask, and the masking parameters for a source. When you assign data masking rules to policies, you can assign multiple source columns to the data masking rules. You can also assign a rule directly to a source column. You can assign data masking rules based on the data type of the source columns.

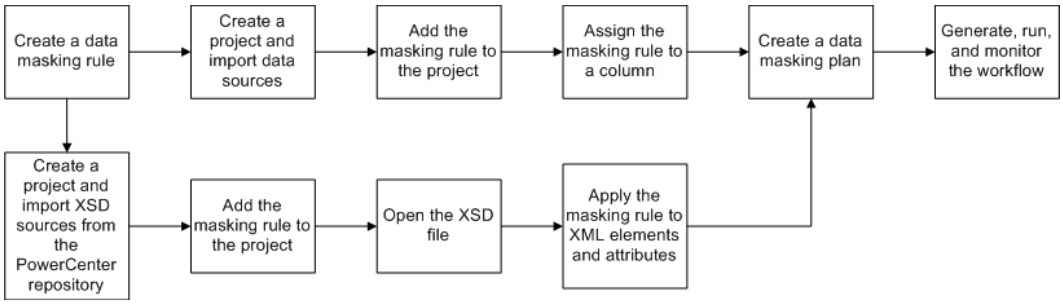
To implement data masking, create a data masking plan and generate a workflow from the plan. A data masking plan can contain policies and rules. In a data masking plan, select a rule that is assigned to a column. Policies and rules define how to mask sensitive and confidential data in a target database. A data masking plan contains at least one rule or one policy.

When you start the workflow, the PowerCenter Integration Service performs the masking operation.

You can apply data masking rules and policies to XML elements and attributes to mask XML data. Open the XSD file from a project and view the details of the XML elements and attributes in the XSD map view and the grid view. You cannot use shuffle masking technique for XSD sources.

Data Masking Task Flow

To implement data masking operations, assign masking rules to columns in a source. Create a plan and add policies and rules to the plan. Generate a workflow from the plan to mask data in a target database.



You can perform the following tasks related to data masking in Test Data Manager:

Create data masking rules

You can create a data masking rule and add the masking rule to a project. You can also add a masking rule to a data domain and add the data domain to a policy. Add the masking rule to a data domain, and add the data domain to policies.

Edit data masking rules

You can edit a data masking rule to change the rule parameters.

Assign data masking rules to columns

Use the masking rules within a project to assign to target columns. You can assign a masking rule or a policy to a target column. You can assign the masking rules to XML elements and attributes if you want to mask data from an XSD source.

Run a plan to mask sensitive data

Create a plan and add data masking components. To mask sensitive data, you must generate and run the plan.

Data Masking Rules

A data masking rule is a data masking technique to mask a specific type of data. You can create a standard rule, advanced rule, or a rule that you import as a mapplet.

A data masking technique defines the logic that masks the data. Masking parameters are options that you configure for a masking technique. For example, you can define different dictionary files for substitution masking rules. Dictionary files contain the sample data for substitution. You might blur output results by different percentages for different columns. Most masking techniques have associated masking parameters.

You can enable users to override masking parameters for a rule. For example, you create a rule with the substitution masking technique to mask column data based on a flat file substitution source. You set the override option for the rule. When a developer assigns this rule to columns in a source, the developer can select a relational database as a substitution source rather than a flat file.

You can assign rules to source columns, data domains, policies, and plans.

Standard Masking Rules

A standard masking rule is a data masking rule that applies a built-in masking technique. A standard masking rule has one input column and one output column.

When you create a standard masking rule, you select the masking technique from a list. You can define one masking technique in a standard masking rule and you can apply the rule to one column.

Test Data Manager has masking techniques that you can select to create masking rules. You can use standard masking techniques based on the source datatype and masking type that you configure for a column. You can restrict the characters in a string to replace and the characters to apply in the mask. When you mask numbers and dates, you can provide a range of numbers for the masked data. You can configure a range that is a fixed or percentage variance from the original number.

Rule Simulation

You can simulate the output of a standard rule to preview the output before you assign the rule to a column.

Use the Rule Simulator to view the output of a standard rule before you assign it to a column or add it to a plan. View the output of the rule and change the rule properties if required before assigning it to a column. You can choose to include data from a connection in the simulation. Alternatively, you can use default sample data or enter up to 100 rows of sample data on which to simulate the rule output. View the original data values and the masked values in the Rule Simulator tab.

The latest simulation configuration details are stored in the browser cache. You can edit the properties of a rule after viewing the simulation results, and run the simulation again on the same data with the updated rule. Clearing the cache deletes the configuration information.

Mapplet Rules

You can create rules from a mapplet. The mapplet contains the logic to mask the input columns and return data to the target columns. When you create a rule from a mapplet, you assign the mapplet column names to input and output columns when you assign the rule to a column in the data source.

Import a mapplet from an XML file that you exported from the Model repository or the PowerCenter repository. The mapplet can contain any passive transformations.

To perform TDM masking operations, you can import PowerCenter mapplets.

A mapplet can contain multiple input and multiple output columns. All columns might not be available in all projects. You must configure one input column and one output column as required columns. The mandatory columns must have source and target assignments when you assign the rule to a column in the data source. Test Data Manager has an interface to assign multiple columns to a rule from a mapplet.

The TDM repository stores the mapplet logic when you import the mapplet. You cannot change the mapplet in Test Data Manager.

Advanced Masking Rules

An advanced masking rule is a combination of masking techniques that mask multiple source columns or a target column based on values of more than one input column.

For example, you can create a full masked name by masking the first name and last name input columns. Define variable columns to contain the masked names. Add an output column that contains a result of an expression that combines the first name and last name variable columns.

Create the following types of columns in an advanced rule:

Input

The source column that you want to mask.

Variable

A column that contains intermediate values in a calculation. The variable column receives a value from an expression or a masking technique. You can configure multiple variable columns in order to combine multiple masking techniques.

Output

The target column that receives the masked value. The output column type contains a masking technique and masking parameters.

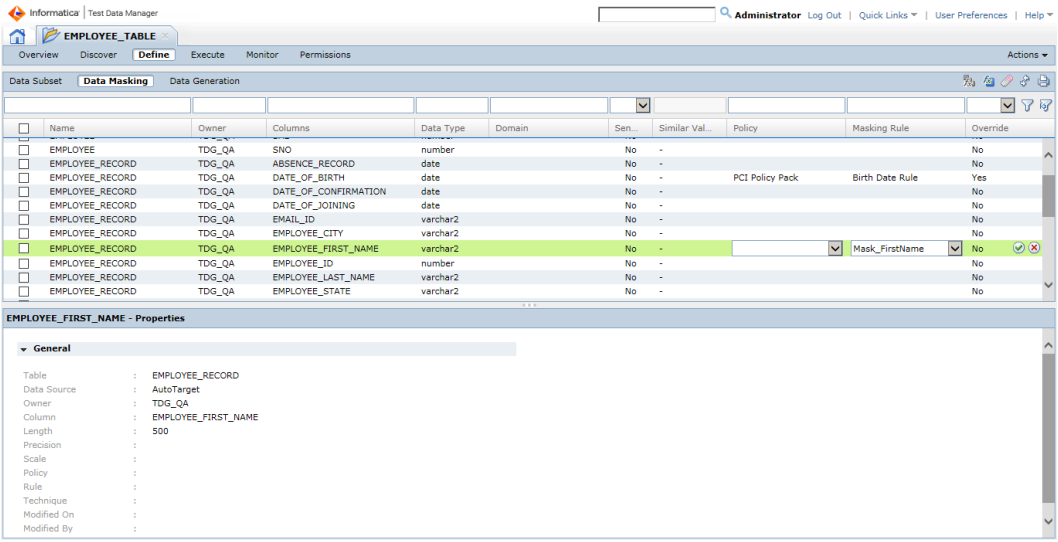
Related Topics:

- "Advanced Masking" on page 148
- "Creating a Data Masking and Data Subset Plan" on page 290
- "Generating a Workflow" on page 293

Masking Rule Assignments

You can assign a data masking rule to a column in the **Define | Data Masking** view. Choose a rule from a list in the **Masking Rule** column in the view. The data domain default rule appears at the top of the list when you click the **Masking Rule** column.

The following image shows the **Define | Data Masking** view in a project:



The following table describes the fields in the **Data Masking** view:

Column	Description
Name	Name of the table.
Owner	Name of the database schema.
Columns	Name of the column to mask.

Column	Description
Data Type	Data type of the column to mask.
Domain	Name of the domain that you assigned to the column either from a data domain discovery or a manual assignment.
Sensitive	Indicates if the column is a sensitive column. Value is Yes or No.
Similar Value Columns	Indicates that the column is configured with other columns in a cascade. The column shows the number of other columns in the cascade.
Policy	Policy name that the column is assigned to.
Masking Rule	The rules to apply to the column. When you click inside the Masking Rule column, you can choose which rule to apply to the column. A preferred rule from a data domain has an asterisk (*) before the name.
Override	Shows the override property status for a rule. If the property is Yes, you can override the rule properties when you assign the rule to a column. When you override the rule parameters for a column the **Override** column value is Yes-Overridden.

Creating and Assigning Data Masking Rules

Create and assign data masking rules to perform data masking operations. Update the masking rule properties, and change the masking rule assignments.

The **Policies** view shows the masking rules in the TDM repository. After you create a masking rule, you can edit and delete a masking rule from the **Policies** view.

Creating a Standard Masking Rule

Create a rule to define a masking technique, the datatype to mask, and masking parameters that define how to apply the technique.

1. To access the **Policies** view, click **Policies**.
2. Click **Actions** > **New** > **Masking Rule**.

   The **Rule Wizard** appears.
3. Enter a name and optional description for the rule.
4. Select the datatype of the column to apply the masking rule to.
5. Select the Standard masking rule.
6. To enable users to override masking parameters for a rule, select the **Override Allowed** option.
7. Click **Next**.

   Note: The **Masking Parameters** dialog box changes based on the **Masking Technique** you select.
8. Enter the masking parameters.
9. Enter the exception handling options. Configure how to handle null or empty spaces. Configure whether to continue processing on error.

10. Click **Finish**.

Previewing Rule Output

Use the Rule Simulator to view the output of a standard masking rule on selected data. You can use data from a connection, use default sample data, or enter sample data on which to view the rule output.

1. To access the **Policies** view, click **Policies**.
2. Click a masking rule name to open the **Rule Properties** page of the masking rule.
3. Click **Rule Simulator** to open the **Rule Simulator** configuration tab.
4. Select the type of configuration from the Source Details list and click **Edit**. Select Default to use default sample data, My Test Data to enter sample data, or Connection to use data from a connection. The **Configure Source Details** page opens.
5. Optional. To use source data from a connection, on the **Configure Source Details** page:
   a. Open the **Connection** tab.
   b. Select the connection and owner from the lists.
   c. Click **Browse** to select the table from a list.
   d. Select the required column from the list of columns.

      By default, up to 10 random columns appear in the list. You can enter the name of a column that you require if it does not appear in the list.
   e. Enter the number of rows to include in the simulation. The default value is 20. The maximum number of rows you can include is 100.
   f. Click **OK**.
6. Optional. To enter sample data:
   a. Enter the data in the data fields on the **Sample Data** tab. Use the buttons to add or delete rows. You can enter a maximum of 100 rows.
   b. Click **OK**.
7. Optional. To use default sample data:
   a. Click the **Copy default data** button on the **Sample Data** tab.
   b. Click **OK**.
8. On the **Rule Simulator** tab, click **Go** to start the simulation.

   The original source values and the masked values appear on the **Rule Simulator** tab.

Creating a Mapplet Masking Rule

You can create a data masking rule from a mapplet. The mapplet contains the logic to mask the source fields.

Export the mapplet to an XML file from the PowerCenter repository or the Model repository before you import it to Test Data Manager.

1. To access the **Policies** view, click **Policies**.
2. Click **Actions** > **New** > **Masking Rule**.

   The **Rule Wizard** appears.
3. Enter a name and optional description for the rule.
4. Choose the data type to mask.

5. Select the Mapplet masking type.
6. Browse for the XML file that contains the PowerCenter mapplet to import.
7. Click **Next**.
8. Select at least one input column and output column as mandatory columns.

   Select the column and click the mandatory column to change the value from `No` to `Yes`.
9. Click **Finish**.

Creating an Advanced Masking Rule

Create an advanced masking rule to combine more than one masking technique or to mask multiple columns.

1. To access the **Policies** view, click **Policies**.
2. Click **Actions** > **New** > **Masking Rule**.

   The **Rule Wizard** appears.
3. Enter a name and optional description for the rule.
4. Select the Advanced masking rule.
5. Click **Next**.
6. In the **Input Columns** section, click **Add New**.

   The **Add Column** dialog box appears.
7. Enter the column properties, such as the name, datatype, precision, and scale. Select whether the column is mandatory to assign to a data source in all projects.

   You cannot enter masking properties for input columns.
8. Click **OK**.
9. To enter more input columns, click **Create Input Column** in the **New Masking Rule** dialog box.
10. In the **Variable Columns** section, click **Add New**.
11. Enter an expression, a masking rule, or a dependent column for each variable column that you define. If you apply a masking rule, configure the input column to create the variable column from.
12. Click **OK**.
13. In the **Output Columns** section, click **Add New**.
14. Enter an expression, a masking rule, or a dependent column for the output column.
15. Click **OK**.

Adding Data Masking Rules to a Project

After you create the data masking rules, you must add the rules to a project. Use the masking rules within the project to mask target columns.

1. Click **Projects**.

   You can see a list of projects.
2. Open the project in which you want to use the rule.

   The project window opens in another tab.
3. Click **Overview** > **Policies**.
4. Click **Actions** > **Add Additional Rules**.

The **Add Additional Rules** dialog box appears.

5. Select the masking rules that you created.
6. Click **OK**.

   The masking rules appear under the **Additional Rules** list.

Assigning a Standard Masking Rule

You can assign a standard masking rule to a target column. Assign masking rules from data domains or policies to one or more columns in the project source that you want to mask.

1. In the project, click **Define | Data Masking** to access the **Data Masking** view.
2. Select a column to assign a masking rule to.
3. If the **Domain** is blank for the column, click the **Policy** column and choose a policy that contains the data masking rule that you want to assign.
4. Click inside the **Rule** column to view the list of available rules.

   The data domain preferred rules appear at the top of the list. The other rules in the policy appear at the bottom of the list.
5. Select a masking rule. If you choose substitution masking rule, you can specify the rule assignment parameters. If you choose mapplet or advanced rule, you can assign the rule columns to the columns in the project.
6. Click **Save** for each column that you update.
7. If you want to update a masking rule assignment, select another masking rule.

   If the masking rule contains assignments, the **Impacted Objects** dialog box appears with the list of affected plans, columns, and data domains.
8. To download the list of affected columns and plans, click **Export**, and save the .csv file.
9. To save the changes, click **Continue**.

   To update the changes in a plan, you must generate and run the plan again.

Assigning a Custom Masking Rule

Mapplet rules can require values from multiple ports. You must map each source input column to a rule input and output port that you configure in the mapplet rule.

1. In the **Define | Data Masking** view, click the **Masking Rule** column for the column that requires a mapplet rule.
2. Select the mapplet rule from the list.

   The **Custom Rule Assignment** dialog box appears with the list of columns in the table.
3. Select a source input column in the left panel and a rule input port in the right panel.

   When you select a source input column, the rule ports with same data type become available for mapping.
4. Click the **Link** icon to create a mapping.

   An arrow appears that links the source port to the rule port.

   Note: To create a mapping, you can also click **Show Simple** and map the ports.
5. If you need to remove the link, select the source input port and the rule input port. Click the **Unlink** icon.

   The arrow between the ports disappears.

6. After you map the rule input ports, click **Next**.
7. Select a rule output port in the left panel and a source input column in the right panel.
8. Click the **Link** icon to create a mapping. To remove a mapping, click the **Unlink** icon.
9. Click **Save**.

Assigning an Advanced Masking Rule

Assign an advanced masking rule to a column and map source input columns to the rule input and output ports. To perform expression cascade, you can add another table, select columns from the table, and map the input columns. An expression cascade joins the two tables to generate a combined masked output.

1. In the **Define | Data Masking** view, click the **Masking Rule** column for the column that requires an advanced rule.
2. Select the advanced rule from the list.

   The **Advanced Rule Assignment** dialog box appears with the list of columns in the table.
3. To select columns from another table, click the **Add Associated Table** icon.

   The **Add Associated Table** dialog box appears.
4. Select an associated table and click **OK**.

   The table name appears with a list of column.
5. Select a source input column in the left panel and a rule input port in the right panel.

   Note: When you select a source input column, the rule ports with same data type become available for mapping.
6. Click the **Link** icon to create a mapping.

   An arrow appears that links the source port to the rule port.

   Note: To create a mapping, you can also click **Show Simple** and map the ports.
7. If you need to remove the link, select the source input port and the rule input port. Click the **Unlink** icon.

   The arrow between the ports disappears.
8. After you map the rule input ports, click **Next**.
9. Provide a join condition for the two source input tables to perform expression cascade.
10. Click **Next**.
11. Select a rule output port in the left panel and a source input column in the right panel.
12. Click the **Link** icon to create a mapping. To remove a mapping, click the **Unlink** icon.
13. Click **Save**.

Applying a Data Masking Rule to XML Elements and Attributes

Apply data masking rules to XML elements and attributes to mask data from an XML source.

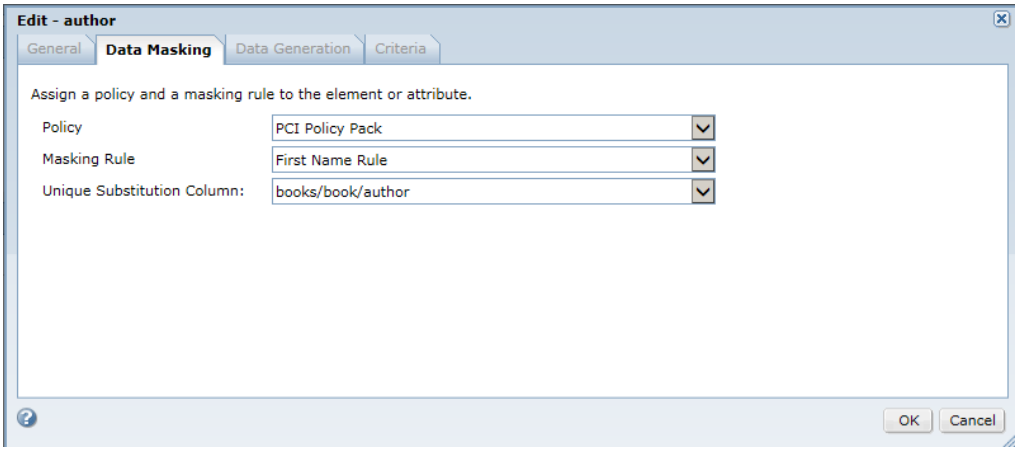
1. In a project, click **Discover** > **Files**.

   A list of XSD files appears in the project.
2. Click the XSD file.

   The XSD file opens in a separate tab.
3. Select the elements and attributes on which you want to apply the masking rule.

4. In the **Details** pane, click **Edit** in the **Data Masking** section.
5. Assign a policy and a data masking rule to the selected element or attribute.

   The following image shows a sample data masking rule assignment to an XML element:



6. To save the assignment, click **OK**.
7. To configure value cascades, click **Set as Similar Value Objects**.

   The **Set as Similar Value Elements and Attributes** dialog box appears.
8. Select the driving object and click **OK**.
9. To assign default rules to multiple objects, click **Default Assignment**.
10. To clear rule assignments, click **Clear Assignment**.

Modifying Data Masking Rules and Assignments

You can update or delete masking rules after you assign the rules to the target columns.

When you update or delete a masking rule, a warning message appears that contains the list of columns, plans, and data domains that contain the rule assignment. You can export the list of affected objects and save them in a .csv file.

If you update a masking rule, you change the properties within the masking rule. The changes do not take effect in the plans that contain the rules. You must fix the changes in the plan, and generate and run the plan again. If you make changes in the name or type of the field of a custom masking rule, the assignments do not contain the changes. You must import the mapplet again.

If you delete masking rules, TDM does not delete the masking rule from the columns, data domains, and plans that contain the rule assignment. To update the changes in a plan, you must generate and run the plan again.

Available Data Masking Rules

The **Data Masking** view lists the rules that you can assign to each column of the data source. You can assign one or more rules for a column if the masking rules are available for the column.

If no rules appear for a column when you click **Masking Rule**, check for the following situations:

- The project has no rules for the column datatype.
- The project does not contain policies with masking rules.
- A data domain that has a preferred rule that you need to use is not in a policy that you assigned to the project.

The PowerCenter Integration Service does not mask integers greater than 28 characters.

If a column has a data domain assignment, the preferred rules for the data domain appear at the top of the list in the **Masking Rule** column. If you forget to add a rule to a data domain, you can assign rules from policy preferred rules. The policy preferred rules appear below the data domain rules in the rules list.

You can apply the default data domain rules to multiple columns at a time. Select multiple columns and click **Rule Assignment**. You can choose to assign the default rules in the **Rule Assignment** dialog box.

You can apply a single masking rule and a policy to multiple columns of similar data type. Select the rule from the **Single Rule Assignment** dialog box. For example, you can search for an ID column, select all the ID columns, and assign a single rule, a policy, or both to the selected columns.

When you select mapplet rules, advanced rules, or any rule that uses multiple columns, you must configure a mapping between the source ports and the rule ports. The Test Data Manager notifies you when you need to configure the source to rule port mapping.

Note: In the **Data Masking** view, you must click the **Save** icon after each time you assign a rule. If you do not save the rule assignment before you assign another rule, the Test Data Manager discards the rule assignment.

Editing a Masking Rule

You can edit a masking rule and update the masking rule properties that you want.

1. To access the **Policies** view, click **Policies**.
2. Click the name of the masking rule that you want to edit.

   The rule opens in a tab.
3. Click **Actions** > **Edit**.
4. Edit the parameters that you want to change and click **OK** to save the rule.

   If the masking rule has assignments, the **Impacted Objects** dialog box appears with the list of affected columns, plans, and data domains.
5. To download the list of affected columns and plans, click **Export**, and save the .csv file.
6. To save the changes, click **Continue** .

   To update the changes in a plan, you must generate and run the plan again.

Copying a Masking Rule

You can create a masking rule by copying a masking rule. When you copy a masking rule, Test Data Manager copies the masking rule properties from the original masking rule to the new masking rule.

1. To access the **Policies** view, click **Policies**.
2. Click a masking rule description to select the masking rule.

   Do not open the masking rule.
3. Click **Actions** > **Duplicate**.

   The **Copy <Masking Rule Name>** dialog box appears.
4. Change the name and description of the masking rule. Click **Save**.

Deleting a Masking Rule

You can delete a masking rule before or after you assign the masking rule to a column.

1. To access the Policies view, click **Policies**.
2. Click the box to the left of the generation rule that you want to delete.
3. Click **Actions** > **Delete**.

   The **Delete Masking Rule** dialog box appears with a warning message. If the masking rule has assignments, the **Impacted Objects** dialog box appears with the list of affected columns, plans, and data domains.
4. To delete the masking rule that do not have assignments, click **OK**.
5. To delete the masking rule that have assignments, click **Continue**. To download the list of affected objects, click **Export**, and save the .csv file.

   To update the changes in a plan, you must generate and run the plan again.

Overriding a Masking Rule

You can override rule parameters after you assign the masking rule to a column. The rule must have the override property enabled.

1. In the **Define | Data Masking** view, select a column that has a masking rule with the override property enabled.

   The **Override** column value is Yes.
2. Click **Actions** > **Override**.

   The **Edit <Rule Type>** dialog box appears and shows the rule parameters.
3. Change the rule parameters and the exception parameters as required.
4. Click **Save**.

   Test Data Manager adds an Overridden flag to the **Override** column when you override the rule.

Assigning Single and Multiple Masking Rules

You can assign multiple masking rules to a column. To assign more than one rule to a column, create a duplicate row and update the rule. You can assign the default data domain rules to multiple columns at the same time. Assign a single masking rule or a policy or both to multiple columns of similar data type.

1. Open the **Define | Data Masking** view for a project.
2. Select a column that you want to assign the rule to. If you want to assign a rule to more than one column, you can select multiple columns.

   Use the checkbox to select the column.
3. Click **Rule Assignment**.

   The **Rule Assignments** dialog box appears.
4. Select the column that you want to assign multiple policies to.

   Use the checkbox to select the column.
5. Click **Copy Rule Assignment**.

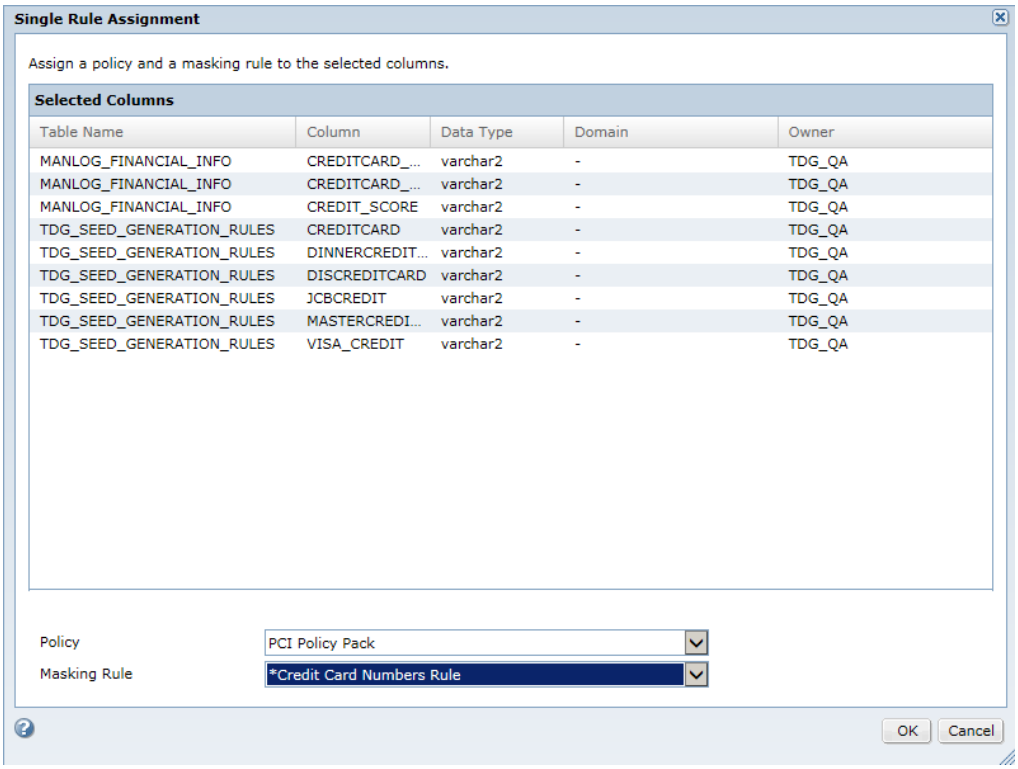
   The Test Data Manager creates a row that is a duplicate of the row you selected.
6. Change the policy to view more preferred masking rules.
7. In the **Masking Rule** column, select a rule from the list.
8. Click **Save**.

   The **Define | Data Masking** view shows two rows for the same column. Each column has a different rule assignment.
9. To assign the default rule to multiple columns, select which columns you want to update with the default values. You can select the checkbox at the top of the dialog box to select all rows.
10. Click **Default Assignments**.

    The Test Data Manager updates each column with the default rule.
11. Click **Save**.
12. To assign a single rule to multiple columns, search for the similar data type columns.
13. Select all the columns that you want to assign a masking rule to.
14. Click **Single Rule Assignment**.

    The **Single Rule Assignment** dialog box appears.
15. Select a policy and a masking rule.

The following image shows the single masking rule assignment to multiple columns:



16. Click **OK**.

    The Test Data Manager updates each column with the assigned masking rule.

Deleting Masking Rule Assignments

You can delete a masking rule assignment to a column.

1. In the **Define | Data Masking** view, select the column that you need to delete the rule assignments for.
2. Click **Clear Assignment**.

   Test Data Manager prompts you to confirm deleting the assignment.
3. To delete the rule assignment from the column, click **OK**.

   If a plan contains the masking rule assignment, the **Impacted Objects** dialog box appears with the list of affected columns, plans, and data domains.
4. To delete the masking rule assignment, click **Continue**. To download the list of affected objects, click **Export**, and save the .csv file.

   To update the changes in a plan, you must generate and run the plan again.

Performing a Data Masking Operation

Perform a data masking operation in Test Data Manager to mask sensitive data.

1. Create a data masking rule.

2. Create a project and import metadata into the project.

   Create different projects to work with XSD sources and relational sources.
3. Add the data masking rule to the project.
4. Assign the masking rule to a target column. If the target is an XML file, apply the masking rule to an XML element or an attribute.
5. Create a plan, add the data masking rule, and configure the source and target connections.
6. Generate and run the workflow.
7. Monitor the progress of the workflow.

Related Topics:

- "Creating a Project" on page 52
- "Importing Data Sources" on page 60
- "Plan Settings" on page 268
- "Creating a Data Masking and Data Subset Plan" on page 290
- "Workflow Generation" on page 293
- "Executing a Workflow" on page 295
- "Viewing the Log Messages" on page 304

Data Masking Components

To perform data masking operations, assign rules to data domains, policies, and columns. Use data domains and data discovery to find columns that you want to mask. Create cascades to mask similar columns.

The following table describes the components that you create to implement data masking operations:

Component	Description
Assignments	The allocation of rules to a column to mask the column data. You assign a rule to a column through either a column assignment or a data domain assignment. A column assignment assigns a rule directly to a column in a source. A data domain assignment assigns one or more rules in a data domain to columns in a source.
Column sensitivity	A sensitive column contains sensitive data. Configure column sensitivity to mark columns that you want to mask.
Data domain	An object that represents the functional meaning of a column based on the column data or the column name. Use a data domain to filter the ports that you want to mask when you assign a rule to columns. Define patterns in the data or patterns in the column names when you configure a data domain.
Plan	Defines data masking operations. A data masking plan indicates whether to mask data in place in the source database or in stream in a target database.
Policy	Defines the data masking rules, the data to mask, and the masking parameters for a source.

Component	Description
Rule	Defines the data masking technique, an optional rule qualifier, and masking parameters. A masking technique defines the logic that is used to mask the data. Masking parameters define how a masking technique in a rule masks source data. You can set an override option in a rule that defines whether users can modify the masking parameters for the rule when they assign the rule to columns in a source.
Value cascade	Masks similar columns across tables. You can identify similar columns in a project and configure them to cascade masking rules. Use cascades when some fields are denormalized across multiple tables.

Masking Components in PowerCenter

When you generate and run a workflow from a data masking plan, the PowerCenter Integration Service creates the masking components that are required to run a data masking workflow. The PowerCenter Integration Service stores these components in the PowerCenter repository.

The PowerCenter Integration Service creates the following components to perform the data masking operations:

- Mappings
- Transformations
- Mapplets
- Sessions and workflows

Important: If you edit the components that are generated from Test Data Manager in the PowerCenter Client, the session might generate unexpected results.

Mapplet Export

To implement PowerCenter masking techniques that are not available in Test Data Manager, you can create and export masking mapplets from PowerCenter to an XML file. You can import mapplets as rules into Test Data Manager.

Include one or more Data Masking transformations in the mapplet. The mapplet can be a single or multiple port mapplet. The transformations must be passive transformations. For a single input and output port mapplet, you can configure the same port as both the input and output port, and you can complete this action for multiple ports at a time when you assign a mapplet that you imported as a rule to columns in a project.

When you assign a mapplet to a columns, assign the source and target columns to the mapplet. The TDM Server applies the PowerCenter transformation within the mapplet to the columns. You can also export rules as PowerCenter mapplets.

CHAPTER 8

Data Masking Techniques and Parameters

This chapter includes the following topics:

- Data Masking Techniques and Parameters Overview, 144
- Data Masking Techniques, 144
- Data Masking Parameters, 145
- Custom Masking, 148
- Advanced Masking, 148
- Credit Card Masking, 150
- Email Masking, 151
- Encryption Masking, 153
- Expression Masking, 155
- IP Address Masking, 156
- Key Masking, 156
- Nullification Masking, 159
- Phone Masking, 159
- Random Masking, 159
- Shuffle Masking, 164
- SIN Masking, 165
- SSN Masking, 165
- Substitution Masking, 166
- URL Masking, 167
- Name Substitution Example , 168
- Shuffle Address Example, 174

Data Masking Techniques and Parameters Overview

A data masking technique is the type of data masking to apply to a selected column. The masking parameters are the options that you configure for the technique.

The type of masking technique that you can apply depends on the datatype of the column that you need to mask. When you choose a masking technique, Test Data Manager displays parameters for the masking technique.

You can restrict the characters in a string to replace and the characters to apply in the mask. You can provide a range of numbers to mask numbers and dates. You can configure a range that is a fixed or percentage variance from the original number.

You can configure different masking parameters for a masking technique and save each configuration as a masking rule. The PowerCenter Integration Service modifies source data based on masking rules that you assign to each column. You can maintain data relationships in the masked data and maintain referential integrity between database tables.

Data Masking Techniques

You can apply masking techniques based on the source datatype that you configure for a column. For example, if the column datatype is numeric, you can define a masked value that is within a fixed or percent variance from the original value.

The PowerCenter Integration Service does not mask integers greater than 28 characters.

The following table describes the masking techniques that you can choose when you define a rule:

Masking Technique	Description
Advanced	Applies masking techniques to multiple input and output columns. You can create an expression to combine multiple columns. You can mask all datatypes.
Credit card	Applies a built-in mask format to disguise credit card numbers. You can mask the string datatype.
Email address	Applies a built-in mask format to disguise email addresses. You can mask the string datatype.
Encryption	Applies encryption to the source data. You can assign an encryption method to mask the data with. You can encrypt string data types.
Expression	Applies an expression to a column to create or disguise data. Use expression masking if the expression references one column. If the expression includes multiple columns, use the calculated masking technique with advanced masking. You can mask all datatypes.
IP address	Applies a built-in mask format to disguise IP addresses. You can mask the string datatype.
Key	Produces deterministic results for the same source data, masking rules, and seed value. You can mask date, numeric, and string datatypes.
Mapplet	Applies masking rules from a PowerCenter mapplet. The mapplet contains the logic to mask the input columns and return data to the target. A mapplet can have multiple input and output columns.

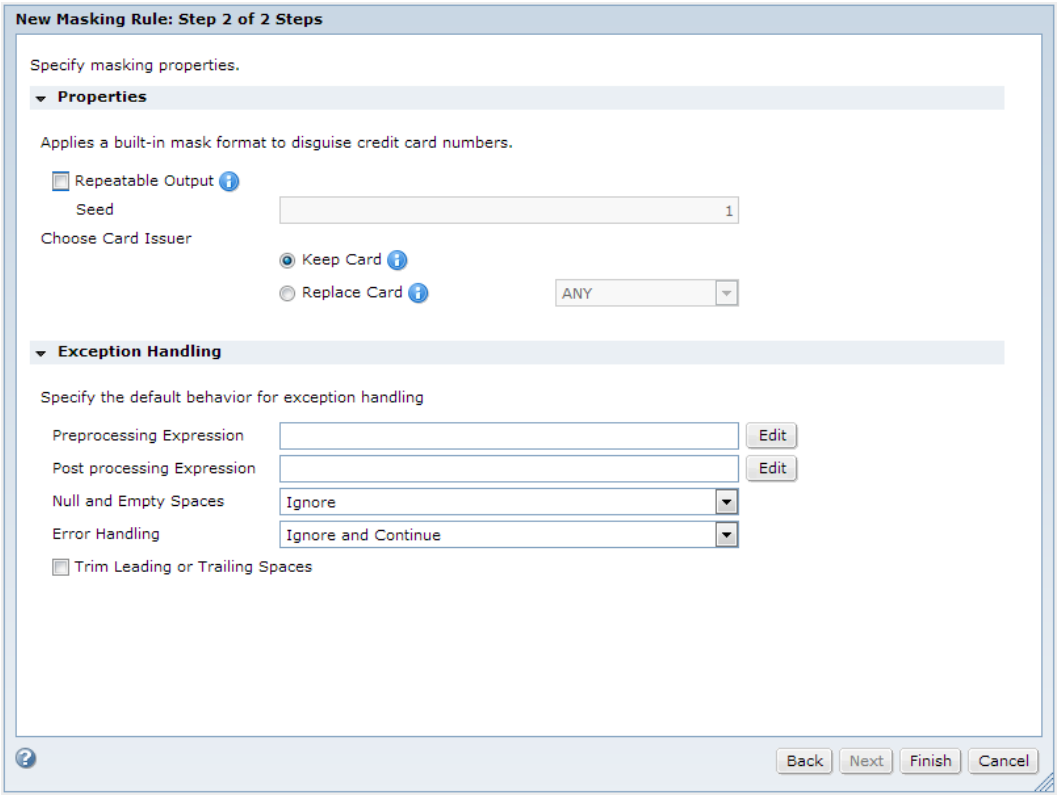
Masking Technique	Description
Nullification	Replaces a column of data with a null value. You can mask all datatypes.
Phone	Applies a built-in mask format to disguise phone numbers. You can mask the string datatype.
Random	Produces random, non-repeatable results for the same source data and masking rules. You can mask date, numeric, and string datatypes.
Shuffle	Applies sensitive column values from one row to another row in the same table. You can restrict which rows to shuffle data from. You can mask date, numeric, and string datatypes.
SIN	Applies a mask to Social Insurance numbers. You can mask a string datatype.
SSN	Applies a built-in mask format to disguise Social Security numbers. You can mask the string datatype.
Substitution	Replaces a column of data with similar but unrelated data from a dictionary. You can mask date, numeric, and string datatypes.
URL	Applies a built-in mask format to disguise URL data. You can mask a string datatype.

Data Masking Parameters

Configure data masking parameters to define how to apply a data masking technique. The parameters that you configure depend on the data type that you need to mask. Some masking techniques are not available for all data types.

You can configure the data masking properties and the exception handling parameters in the **New Masking Rule** wizard.

The following image shows data masking parameters that appear when you configure a Credit Card data masking rule:



Repeatable Output

Repeatable data masking output returns deterministic values. Use repeatable masking when you generate a data masking workflow more than once and you need to return the same masked values each time it runs.

Configure repeatable output if you have the same value in multiple source tables and you want to return the masked value in all of the target tables. The tables in the target database receive consistent masked values.

For example, customer John Smith has two account numbers, 1234 and 5678. The account numbers are in multiple tables. The PowerCenter Integration Service masks John Smith as Frank Martinez in all the tables. It always masks account number 1234 as 6549 and account number 5678 as 3214.

You can enter a seed value when you configure repeatable output. You can configure a dictionary file with replacement data values for substitution masking. When you configure repeatable output, TDM returns the same value from the dictionary whenever a specific value appears in the source data.

Seed

Apply a seed value to create repeatable output for data masking output. The seed value is a starting point for generating masked values.

You can define a seed value from 1 through 999. The default seed value is 1. Apply the same seed value to a column to return the same masked data values in different source data. For example, you have the same `Cust_ID` column in four tables. If you want all of them to output the same masked values, you can apply the same seed value when you mask each column.

You can create a variable for the seed value in a parameter file. When you configure repeatable output in a masking rule, enter the name of the variable instead of a seed value. You can then change the seed value in the parameter file if you need to change the seed value, instead of editing the masking rule.

The TDM administrator can also set a central seed value for all data masking components with repeatable output. The central seed value overrides any other seed value.

Exception Handling

Data masking exception handling defines options for handling nulls, blanks, empty strings, and errors in source data. You can configure exception handling for each data masking rule that you create. You can specify preprocessing and post processing expression parameters to apply changes before and after masking the data.

The following table describes the exception handling options:

Option	Description
Preprocessing Expression	Optional. Expression to define changes to make to the data before masking. Click **Edit** to configure the preprocessing expression.
Post processing Expression	Optional. Expression to define changes to make to the masked data before saving the data to the target. Click **Edit** to configure the post processing expression.
Null and Empty Spaces	The default behavior for handling null values or empty columns in the source data. Choose one of the following options: - Constant. Mask the data with a constant value. Enter the value to use. - Default server value. Mask the source column with the default value from the defaultValue.xml file. - Log error and continue. Log an error in the PowerCenter session log and continue processing. - Treat as value. Treat null values or spaces as a valid source value. Mask the space or null value with a valid value. - Ignore. Do not mask the null or empty space. **Note:** For columns with dependent masking, there are no separate rules to handle null and empty spaces. TDM masks the dependent column based on the rules and values of the column that it depends on.
Error Handling	The default behavior for handling errors in the source data. Choose one of the following options: - Constant. Mask the data with a constant value. Enter the value to use. - Default server value. Mask the source column with the default value from the defaultValue.xml file - Log exception and continue. Log an exception in the PowerCenter session log and continue processing. - Ignore and continue. Do not mask the null or empty space. - Error. Log an error in the PowerCenter session log and stop processing.
Trim Leading or Trailing Spaces	Trims the leading and trailing spaces from source data. When you enable this option the following source fields are the same: " Jones", "Jones", "Jones " .

Custom Masking

Use custom masking when you want to apply multiple masking techniques to a column or when you want to configure multiple input and output columns in the same masking rule. You can mask all data types.

To create a custom masking rule, you must import the PowerCenter mapplets. Set the mapplet input columns and output columns as required. The mapplet must contain at least one input and one output column.

When you assign the custom masking rule to columns, you must map the source columns to the columns in the custom masking rule.

Custom Masking Parameters

Configure input and output columns that you import from a mapplet when you create a custom masking rule.

The following table describes the general properties that you can configure for input and output columns:

Parameter	Description
Column Name	The name of an input or an output column within a mapplet. When you assign the rule to a column, you map the column names in the rule to column names in the database.
Column Type	The column type. The mapplet can contain the following types of columns: - Input. Receives the input data. - Output. Returns the output data.
Data Type	The data type of the column.
Precision	The precision for the column. The maximum number of digits or the maximum number of characters that the column can accommodate. For example, 874.560 has a precision of 6.
Mandatory	Indicates if you must assign the column to a table column in the project. Applies to input and output columns. You must set at least one input and one output column as required.
Group Name	Specifies the group to which the columns belong. The group name can be input, output, or any other name that you provide when you create a mapplet.

Advanced Masking

Use advanced masking when you want to apply multiple masking techniques to a column or when you want to configure multiple input and output columns in the same masking rule. You can mask all datatypes.

When you configure advanced masking, you configure the input columns, output columns, and variable columns. Variable columns are work fields that you can define to temporarily store data.

When you create the columns in the masking rule, the column names do not need to be the same as the column names in the source. When you assign the masking rule to columns, you must map the source columns to columns in the advanced masking rule.

Related Topics:

- "Advanced Masking Rules" on page 129
- "Creating a Data Masking and Data Subset Plan" on page 290

- "Generating a Workflow" on page 293

Advanced Masking Parameters

Configure parameters for each column that you create in an advanced masking rule.

The following table describes the general properties that you can configure for input, output, and variable columns:

Parameter	Description
Column Name	The name of an input, output, or variable column. Enter any name. The name does not have to match the name of a column in the source. When you assign the rule to source data in a project, you map the column names in the rule to column names in the database.
Column Type	The column type. You can configure the following types of columns: - Input. Receives the source data. - Variable. A temporary column that contains intermediate values. You can apply masking rules to variable column values in order to mask data before returning data to output columns. - Output. Returns the output data. You can apply an expression or a masking rule to variable column data and return the data in the output column.
Datatype	The datatype of the column.
Precision	The precision for the column. The maximum number of digits or the maximum number of characters that the column can accommodate. For example, 798.650 has a precision of 6.
Scale	Number of digits to the right of the decimal point in a number.
Mandatory	Indicates if you must assign the column to a column in the source. Applies to input and output columns.

The following table describes the masking properties that you can configure for variable columns and output columns:

Parameter	Description
Expression	An expression to apply to the variable column. You can create the expression in the Expression Builder.
Masking Rule	Applies a masking rule to the input column and writes the results in the variable column. You can enter the following parameters. - Condition. Defines whether an input column should be masked or not. If the condition is true, the PowerCenter Integration Service masks the column. - Rule. The data masking rule to apply to the input column. - Override properties. You can change the data masking rule parameters if the rule owner enabled the rule to be overridden. - Input column. The name of the input column to apply the masking rule to. Select an input column from the columns that you added to the rule.

Parameter	Description
Condition Input	Applies an expression to the Output column only. Select a condition from the list. If you applied conditions on the input column, you can use it for the output column.
Dependent	Applies dependent masking. Dependent masking replaces the value of a column based on the values returned from a dictionary row for another column. You must define substitution masking for another column before configuring a column for dependent masking. Enter the following parameter: - Input column. The name of the input column that you configured for substitution masking. - Dictionary column. Choose the dictionary column to replace the dependent column with.

Advanced Masking Example

You can create an expression to combine multiple columns in an advanced masking rule.

Create an expression in the **Expression Builder**. Select columns, functions, variables, and operators to build expressions. The expression can reference input columns and variable columns.

For example, you have a CUSTOMERS table that contains first and last names. You want to mask the names and combine the masked values into a masked full name.

The following table shows the columns that you create in an the advanced masking rule to combine the masked names:

Column Name	Column Type	Masking Technique	Expression
FIRST_NAME	INPUT	–	–
LAST_NAME	INPUT	–	–
FIRST_MASKED	VARIABLE	Substitution on FIRST_NAME	_
LAST_MASKED	VARIABLE	Substitution on LAST_NAME	_
FULL_NAME	OUTPUT	Expression	FIRST_MASKED || ' ' || LAST_MASKED

Mask the FIRST_NAME and LAST_NAME with Substitution. Return FULL_NAME using an expression that combines the FIRST_MASKED and LAST_MASKED columns.

When you create expressions in the Expression Builder, use the point-and-click interface to minimize errors. Verify that the expression returns a value that matches the output column data type.

Credit Card Masking

Credit card masking applies a built-in mask format to disguise credit card numbers. The PowerCenter Integration Service creates a masked number that has a valid checksum. You can choose from multiple credit card number formats. Mask the string datatype with credit card masking.

The PowerCenter Integration Service generates a logically valid credit card number when it masks a credit card number. The length of the source credit card number must be between 13 to 19 digits. The input credit card number must have a valid checksum based on credit card industry rules.

You can apply repeatable masking to credit card numbers. Each credit card type has a different format. You can choose to keep the same credit card type in the masked credit card number, or you can change the credit card type. For example, if the source credit card number is a Visa credit card number, you can configure the rule to return a Visa credit card number. Or, you can configure the rule to return a credit card number that is a different credit card type.

The source credit card number can contain numbers, spaces, and hyphens. If the credit card has incorrect characters or is the wrong length, the Integration Service writes an error to the session log. The PowerCenter Integration Service applies a default credit card number mask when the source data is not valid.

The PowerCenter Integration Service does not mask the six-digit Bank Identification Number (BIN) at the beginning of the number. For example, the PowerCenter Integration Service might mask the credit card number `4539 1596 8210 2773` as `4539 1516 0556 7067`.

Credit Card Masking Parameters

The following table describes the parameters that you can configure for credit card masking:

Parameter	Description
Repeatable	Returns the same masked value when you generate a workflow multiple times, or when you generate masked values for a column that is in multiple tables.
Seed	Starting point for creating repeatable output. Enter a number between 1 and 999. Default seed value is 1.
Keep card issuer	The PowerCenter Integration Service returns the same credit card type for the masked credit card. For example, if the source credit card is a Visa card, generate a masked credit card number that is the Visa format.
Replace card issuer	Replaces the source credit card type with another credit card type. When you enable replace card issuer, select which type of credit card to replace it with. You can choose credit cards such as AMEX, VISA, JCB, and MASTERCARD. Default is ANY.

Email Masking

Email masking generates realistic email addresses. You can apply a mask from the masked values of a first and last name. You can apply a constant domain name or create an expression to define the domain. Mask the string datatype with email masking.

You can configure email masking to mask email addresses the same way each time the same user appears in the source data. You can configure an email address from the masked first and last name columns. You can configure email masking as repeatable between workflows, or you can configure email masking to be repeatable in one workflow.

Note: The PowerCenter Integration Service always returns ASCII characters for an email address.

Email Masking Parameters

When you configure email masking, you can configure parameters to mask the user name and the domain name in the email address. You can specify dictionary files that contain the user names and the domain names. You can define an expression that combines names to create realistic email user names.

Email Masking General Parameters

The following table describes the parameters that define how to mask an email address:

Parameter	Description
Repeatable output	Returns the same masked value when you generate a workflow multiple times, or when you generate masked values for a column that is in multiple tables.
Seed	Starting point for creating repeatable output. Enter a number between 1 and 999. Default is 1.
Standard	Replaces an email string with characters. The standard email masking technique does not create realistic domain or user names.
Advanced	Replaces an email string with name and domain values from a dictionary or mapping.
Columns to form email	Indicates where to retrieve the columns to form the email address. Choose one of the following options: - From mapping. Do not use a dictionary for user names. Assign the column names when you assign the rule to source columns. - From dictionary. Mask user names from a dictionary file.

User Name Masking Parameters

The following table describes the parameters that define how to mask the user name in the email address:

Parameter	Description
Dictionary for columns	The dictionary to use when you choose to form an email from a dictionary column. Choose a dictionary that contains first and last names.
Expression	Defines an expression when you select to create an email from a mapping. You can create an expression to define the user name from parts of the first and last names. The column names in the expression do not have to match the source column names. Assign columns to the expression when you assign rules to columns.
First name column	The dictionary column that contains the first name of the email address.
First name length	Length for the first name. For example, you might choose the first letter of the name, or the first 4 letters of the name. Default is 10.
Delimiter	The delimiter between the first and last names of the email address. Choose a period, underscore, or hyphen.
Last name column	Dictionary column that contains the last name of the email address.
Last name length	The length for the last name. For example, you might choose the first 6 characters of the last name. Default is 10.

Domain Name Masking Parameters

The following table describes the parameters that define how to mask the domain name:

Parameter	Description
Constant	A constant value to mask the email address with. Each email address receives the same domain name.
Random	Indicates whether to use a flat file or relational dictionary.
Domain lookup dictionary	The dictionary file to use from the imported sources.
Domain name column	The column that contains a dictionary name in the domain lookup dictionary.

Encryption Masking

Encryption masking applies encryption algorithms to mask source data.

Mask string data types with encryption masking. You can choose the algorithm to encrypt the data.

Select one of the following encryption types:

- Standard Encryption. To configure standard encryption masking, enter an encryption key that is 16 characters or less. Select from the following encryption algorithms:

  AES

  Advanced Encryption Standard with 128-bit encoding.

  CRC

  Cyclic Redundancy Check. Finds data transmission errors or verifies that data is not modified. Computes a checksum.

  MD5

  MD5 Message-Digest Algorithm. One-way cryptographic hash function with a 128-bit hash value.

- Format Preserving Encryption. You can choose to preserve the format and length of the source data or the length of the source data. You can also choose to change the format and length of the source data after encryption.
  You can choose characters that you do not want to encrypt.

  After you encrypt the source data, you can also decrypt it to get back the original data. To decrypt the data, you must run a plan that uses the same pass phrase and the same format preserving encryption rule configuration that you used to encrypt the source data.

  Note: If the source data contains UTF-8 four byte characters, you cannot use format preserving encryption to mask the data.

  Select one of the following encryption options:

  Preserve Format and Metadata

  Use the Preserve Format and Metadata encryption option to preserve the format and the length of the source data. When you choose to preserve format and metadata, all uppercase characters are replaced with uppercase characters, lowercase characters are replaced with lowercase characters,

numbers are replaced with numbers, and special characters are replaced with special characters after encryption. For example, an email address Abc123@xyz.com might become Mpz849#dje!kuw. In this example, if you configure "@" and "." characters as Do Not Encrypt Characters, the email might become Mpz849@dje.kuw.

Preserve Metadata

Use the Preserve Metadata encryption option to preserve the length of the source data. When you choose to preserve metadata, the length of the data remains the same after encryption. For example, a first name Alexender might become jl6#HB91v, where the length remains the same as in the source data.

Change Metadata

Use the Change Metadata encryption option to change the length of the source data after encryption. When you choose to change metadata, the encrypted data does not retain the length and format of the source data. For example, a city name London might become Xuep@8f5, fmch529, or 6ky#ke33h*we.

Note: Before you use the Change Metadata encryption option, you must change the precision of the column you want to apply encryption on in the database.

Use the following formula to calculate the precision and round up the value to the next higher integer:

```
Required Precision = (1.33*Original Precision)+24
```

After you change the column precision in the database, you must update the column precision in TDM. To update the column precision you can either reimport the metadata from the updated database, or manually change the column precision in Test Data Manager.

Encryption Masking Parameters

You can configure parameters for the encryption masking technique.

The following table describes the parameters that you can configure for encryption masking:

Parameter	Description
Encryption Type	The type of encryption. Select Standard Encryption or Format Preserving Encryption.
Encryption Technique	The technique to encrypt or decrypt the data.
Encryption Key	Input key for the standard encryption technique. Enter an input key that is 16 characters or less.
Publish Rule	Choose to configure the format preserving encryption rule as read-only. Only domain administrators can make a read-only rule editable or delete a read-only rule.
Do Not Encrypt Characters	Specify characters that you do not want to encrypt. To delete a character, click the **X** icon that displays with the character. To use the keyboard to delete a character, press **Tab** or **Shift+Tab** to select the character, and then press **Delete** from the keyboard. **Note:** If you add a space character at the beginning or at the end, TDM does not encrypt spaces in the source data but does not show the space character in the **Do Not Encrypt Characters** field.

Expression Masking

Expression masking applies an expression to a column to mask or change data. Mask all data types with expression masking.

To configure expression masking, you create an expression in the **Expression Builder**.

Select column, functions, variables, and operators to build expressions. The expression can reference input columns and output columns.

In expression masking, you can append additional data to the source column data. For example, you want to mask the first names. The source has the FirstName column and you want to concatenate the values in the FirstName column with the string ABC. When you configure the data masking rule, select the FirstName column and enter the following expression in the expression editor.

```
CONCAT(FirstName,'ABC')
```

When you configure expression masking for a column, the column name appears as the expression by default.

Select functions, columns, variables, and operators from the point-and-click interface to minimize errors when you build expressions.

When you create an expression, verify that the expression returns a value that matches the column data type. The PowerCenter Integration Service returns zero if the return value does not match a numeric column. It returns NULL if the return value does not match a string column.

For information about expression syntax, see the *Informatica Transformation Language Reference*.

Expression Masking Parameters

You can configure parameters for the expression masking technique.

The following table describes the parameters that you can configure for expression masking:

Parameter	Description
Repeatable Output	Determines if the masked value should persist for a given column value. You must enter a seed value when you enable repeatable output.
Alphanumeric Seed	The alphanumeric seed is a key that allows multiple data masking rules to generate the same masked values from the same source values. Define the same seed in each data masking rule that requires the same results for a column. The seed can be any text.
Expression	Accepts PowerCenter expressions and performs a calculation based on values within a single row.

Rules and Guidelines for Expression Masking

Use the following rules and guidelines for expression masking:

- You cannot use the output from an expression as input to another expression. If you manually add the output column name to the expression, you might get unexpected results.
- Select functions, columns, variables, and operators from the point-and-click interface to minimize errors when you build expressions.
- If the data masking rule is configured for repeatable masking, specify the storage table in the data masking plan. If the storage table does not exist, the data masking operation fails.

IP Address Masking

IP address masking applies a built-in mask format to change IP addresses. Mask string datatypes with IP address masking.

Use IP address masking to mask data with the string datatype.

IP masking splits an IP address into four numbers, separated by periods. The first number is the network. The PowerCenter Integration Service masks the network number within the network range.

The PowerCenter Integration Service does not mask the class and private network address. The PowerCenter Integration Service masks a Class A IP address as a Class A IP Address and a 10.x.x.x address as a 10.x.x.x address.

For example, the PowerCenter Integration Service can mask `11.12.23.34` as `75.32.42.52`, and `10.23.24.32` as `10.61.74.84`.

Key Masking

Key masking produces deterministic results for the same source data, masking rules, and seed value. Mask date, numeric, and string datatypes with key masking.

The following table describes the parameters that you can configure for key masking:

Parameter	Description
Seed	A start number that enables the PowerCenter Integration Service to return deterministic data. You can mask the date, numeric, and string datatypes.
Mask Format	The type of character to substitute for each character in the input data. You can limit each character to an alphabetic, numeric, or alphanumeric character type. You can mask the string datatype.
Source String Characters	The characters in the source string that you want to mask. You can mask the string datatype.
Result String Replacement Characters	Substitutes the characters in the target string. You can mask the string datatype.

Mask Format

Configure a mask format to limit each character in the output column to an alphabetic, numeric, or alphanumeric character.

If you do not define a mask format, the PowerCenter Integration Service replaces each source character with any character. If the mask format is longer than the input string, the PowerCenter Integration Service ignores the extra characters in the mask format. If the mask format is shorter than the source string, the PowerCenter Integration Service does not mask the characters at the end of the source string.

Note: The mask format contains uppercase characters. When you enter a lowercase mask character, Test Data Manager converts the character to uppercase.

The following table describes mask format characters:

Character	Description
A	Alphabetical characters. For example, ASCII characters a to z and A to Z.
D	Digits. From 0 through 9.
N	Alphanumeric characters. For example, ASCII characters a to z, A to Z, and 0-9.
X	Any character. For example, alphanumeric or symbol.
+	No masking.
R	Remaining characters. R specifies that the remaining characters in the string can be any character type. R must appear as the last character of the mask.

Source String Characters

Configure source string characters to choose the characters that you want to mask.

For example, if you set the number sign (#) as a source string character, it is masked every time it occurs in the input data. The position of the characters in the source string does not matter, and you can configure any number of characters. If you do not configure source string characters, the masking replaces all the source characters in the column.

The source characters are case sensitive. The PowerCenter Integration Service does not always return unique data if the number of source string characters is fewer than the number of result string characters.

The following table describes the options that you can configure for source string characters:

Option	Description
Mask Only	Masks characters in the source that you configure as source string characters. For example, if you enter A and b as source string characters, every instance of A and b in the source data will change. A source character that is not an A or b will not change.
Mask all except	Masks all characters in the source except for source string characters. For example, if you enter "-" as the source string character, every character except for "-" will change.

Result String Replacement Characters

Configure result string replacement characters to specify masking output.

The PowerCenter Integration Service replaces characters in the source string with the result string replacement characters. For example, enter the following characters to configure each mask to contain uppercase alphabetic characters A through F:

```
ABCDEF
```

To avoid generating the same output for different input values, configure a wide range of substitute characters, or mask only a few source characters. The position of each character in the string does not matter.

The following table describes the options for result string replacement characters:

Option	Description
Use only	Masks the source with only the characters you define as result string replacement characters. For example, if you enter the characters A, B, and c, the masking replaces every character in the source column with an A, B, or c. The word "horse" might be replaced with `BAcBA`.
Use all except	Masks the source with any characters except the characters you define as result string replacement characters. For example, if you enter A, B, and c result string replacement characters, the masked data never has the characters A, B, or c.

Date Key Masking

You can configure key masking with dates to generate deterministic output.

You can change the seed to match the seed value for another column to return repeatable datetime values between the columns.

The PowerCenter Integration Service can mask dates between 1753 and 2400 with key masking. The PowerCenter Integration Service always generates valid dates. If the source year is in a leap year, the PowerCenter Integration Service returns a year that is also a leap year. If the source month contains 31 days, the PowerCenter Integration Service returns a month that has 31 days. If the source month is February, the PowerCenter Integration Service returns "February."

Numeric Key Masking Parameters

You can configure key masking for numeric values and generate deterministic output.

When you configure a column for numeric key masking, you can select a seed value for the column. When the PowerCenter Integration Service masks the source data, it applies a masking algorithm that requires the seed.

You can change the seed value for a column to produce repeatable results if the same source value occurs in a different column. Configure repeatable results when you want to maintain a primary key-foreign key relationship between two tables. In each rule, enter the same seed value for the primary-key column as the seed value for the foreign-key column. The PowerCenter Integration Service produces deterministic results for the same numeric values. The referential integrity is maintained between the tables.

String Key Masking Parameters

Configure string key masking to mask all or part of a string. To limit the masking output to certain characters, specify a mask format and result string replacement characters. If you need repeatable output, specify a seed value.

The following table describes the masking parameters that you can configure for key masking string values:

Parameter	Description
Seed	A start number that enables the PowerCenter Integration Service to return deterministic data. Select a seed value between 1 and 1,000. Apply the same seed value to a column to return the same masked data values in different source data.
Mask Format	The type of character to substitute for each character in the input data. You can limit each character to an alphabetic, numeric, or alphanumeric character type.

Parameter	Description
Source String Characters	The characters in the source string that you want to mask. For example, mask the number sign (#) character whenever it occurs in the input data. Leave this field blank to mask all the input characters. The PowerCenter Integration Service does not always return unique data if the number of source string characters is less than the number of result string characters.
Result String Replacement Characters	Substitutes the characters in the target string. For example, enter the following characters to configure each mask to contain uppercase alphabetic characters A through F: `ABCDEF`

Nullification Masking

Nullification masking replaces a column of data with a null value. Use nullification masking to mask binary, date, numeric, or string data.

Nullification masking has no parameters.

Phone Masking

Phone masking applies a built-in masking format to change phone number data. Mask string datatypes with phone masking.

Phone masking does not change the format of the original phone number. For example, phone masking can mask the phone number `(408)382 0658` as `(408)256 3106`.

The source data can contain numbers, spaces, hyphens, and parentheses. Phone masking does not mask alphabetic or special characters.

Random Masking

Random masking produces random, non-repeatable results for the same source data and masking rules.

Random masking does not require a seed value. The results of random masking are non-deterministic. Use random masking to mask date, numeric, and string data types.

The following table describes the options that you can configure for random masking:

Option	Description
Range	A range of output values. The PowerCenter Integration Service returns data between the minimum and maximum values. You can configure a range for date, numeric and string data types.
Blurring	A range of output values with a fixed or percent variance from the source data. Returns data that is close to the value of the source data. You can configure blurring for date and numeric data types.
Mask Format	The type of character to substitute for each character in the input data. You can limit each character to an alphabetic, numeric, or alphanumeric character type. You can configure a mask format for the string data type.
Source String Characters	The characters in the source string that you want to mask. You can configure source string characters for the string data type.
Result String Replacement Characters	Substitutes the characters in the target string. You can configure replacement characters for the string data type.

Range Masking

Configure a range to define an output range for numeric, date, or string data.

When you define a range for numeric or date values, the PowerCenter Integration Service masks the source data with a value between the minimum and maximum values. When you configure a range for a string, you configure a range of string lengths.

Note: When you configure date random masking, the maximum datetime must be later than the minimum datetime.

Blurring

Configure blurring to return a random value that is close to the original value. For random masking of datetime or numeric data, blurring creates an output value within a fixed or percent variance from the source data value.

Date Blurring

To blur a datetime source value, select a unit of time to blur, a high bound, and a low bound. You can select year, month, day, or hour as the unit of time. By default, the blur unit is year.

For example, to restrict the masked date to a date within two years of the source date, select year as the unit. Enter two as the low and high bound. If a source date is 02 February, 2006, the PowerCenter Integration Service returns a date between 02 February, 2004 and 02 February, 2008.

Numeric Blurring

To blur a numeric source value, select a fixed or percent variance, a high bound, and a low bound. The high and low bounds must be greater than or equal to zero.

The following table lists the masking results for blurring range values when the input source value is 66:

Blurring Type	Low	High	Result
Fixed	0	10	Between 66 and 76
Fixed	10	0	Between 56 and 66
Fixed	10	10	Between 56 and 76
Percent	0	50	Between 66 and 99
Percent	50	0	Between 33 and 66
Percent	50	50	Between 33 and 99

Mask Format

Configure a mask format to limit each character in the output column to an alphabetic, numeric, or alphanumeric character.

Note: The mask format contains uppercase characters. When you enter a lowercase mask character, Test Data Manager converts the character to uppercase.

The following table describes mask format characters:

Character	Description
A	Alphabetical characters. For example, ASCII characters a to z and A to Z.
D	Digits. From 0 through 9.
N	Alphanumeric characters. For example, ASCII characters a to z, A to Z, and 0-9.
X	Any character. For example, alphanumeric or symbol.
+	No masking.
R	Remaining characters. R specifies that the remaining characters in the string can be any character type. R must appear as the last character of the mask.

If you do not define a mask format, the PowerCenter Integration Service replaces each source character with any character. If the mask format is longer than the input string, the PowerCenter Integration Service ignores the extra characters in the mask format. If the mask format is shorter than the source string, the PowerCenter Integration Service does not mask the characters at the end of the source string.

Source String Characters

Source string characters are characters that you want to mask in the source. Configure source string characters if you want to mask a few of the characters in the input string.

For example, if you set the number sign (#) as a source string character, it is masked every time it occurs in the input data. The position of the characters in the source string does not matter, and you can configure any number of characters. If you do not configure source string characters, the masking replaces all the source characters in the column.

The source characters are case sensitive. The PowerCenter Integration Service does not always return unique data if the number of source string characters is fewer than the number of result string characters.

The following table describes the options that you can configure for source string characters:

Option	Description
Mask Only	Masks characters in the source that you configure as source string characters. For example, if you enter A and b as source string characters, every instance of A and b in the source data will change. A source character that is not an A or b will not change.
Mask all except	Masks all characters in the source except for source string characters. For example, if you enter "-" as the source string character, every character except for "-" will change.

Result String Replacement Characters

Result string replacement characters are a set of characters that the PowerCenter Integration Service can use to mask to the source data. You can configure the masking rule to mask the source only from the set of characters, or you can configure the masking rule to mask the source with any character except the result string replacement characters.

The PowerCenter Integration Service replaces characters in the source string with the result string replacement characters. For example, enter the following characters to configure each mask to contain uppercase alphabetic characters A through F:

```
ABCDEF
```

To avoid generating the same output for different input values, configure a wide range of substitute characters, or mask only a few source characters. The position of each character in the string does not matter.

The following table describes the options for result string replacement characters:

Option	Description
Use only	Masks the source with only the characters you define as result string replacement characters. For example, if you enter the characters A, B, and c, the masking replaces every character in the source column with an A, B, or c. The word "horse" might be replaced with `BAcBA`.
Use all except	Masks the source with any characters except the characters you define as result string replacement characters. For example, if you enter A, B, and c result string replacement characters, the masked data never has the characters A, B, or c.

Date Random Masking Parameters

To mask datetime values with random masking, either configure a range of output dates or choose a variance.

When you configure a variance, choose a part of the date to blur. Choose the year, month, day, hour, minute, or second. The PowerCenter Integration Service returns a date that is within the range you configure.

The following table describes the parameters that you can configure for random masking of datetime values:

Parameter	Description
Range	The minimum and maximum values to return for the selected datetime value. The date range is a fixed variance.
Blurring	Masks a date based on a variance that you apply to a unit of the date. The PowerCenter Integration Service returns a date that is within the variance. You can blur the year, month, day, or hour. Choose a low and high variance to apply.

Numeric Random Masking Parameters

When you mask numeric data, you can configure a range of output values for a column.

The PowerCenter Integration Service returns a value between the minimum and maximum values of the range depending on column precision. To define the range, configure the minimum and maximum ranges or a blurring range based on a variance from the original source value.

The following table describes the parameters that you can configure for random masking of numeric data:

Parameter	Description
Range	A range of output values. The PowerCenter Integration Service returns numeric data between the minimum and maximum values.
Blurring Range	A range of output values that are within a fixed variance or a percent variance of the source data. The PowerCenter Integration Service returns numeric data that is close to the value of the source data. You can configure a range and a blurring range.

String Random Masking Parameters

Configure random masking to generate random output for string columns.

To configure limitations for each character in the output string, configure a mask format. Configure filter characters to define which source characters to mask and the characters to mask them with.

The following table describes the parameters that you can configure for random masking of string columns:

Parameter	Description
Range	The minimum and maximum string length. The PowerCenter Integration Service returns a string of random characters between the minimum and maximum string length.
Mask Format	The type of character to substitute for each character in the input data. You can limit each character to an alphabetic, numeric, or alphanumeric character type.
Source String Characters	The characters in the source string that you want to mask.
Result String Replacement Characters	Substitutes the characters in the target string.

Shuffle Masking

Shuffle masking masks the data in a column with data from the same column in another row of the table. Shuffle masking switches all the values for a column in a file or database table. You can restrict which values to shuffle based on a lookup condition or a constraint. Mask date, numeric, and string data types with shuffle masking.

For example, you might want to switch the first name values from one customer to another customer in a table. The table includes the following rows:

```
100 Tom Bender
101 Sue Slade
102 Bob Bold
103 Eli Jones
```

When you apply shuffle masking, the rows contain the following data:

```
100 Bob Bender
101 Eli Slade
102 Tom Bold
103 Sue Jones
```

You can configure shuffle masking to shuffle data randomly or you can configure shuffle masking to return repeatable results.

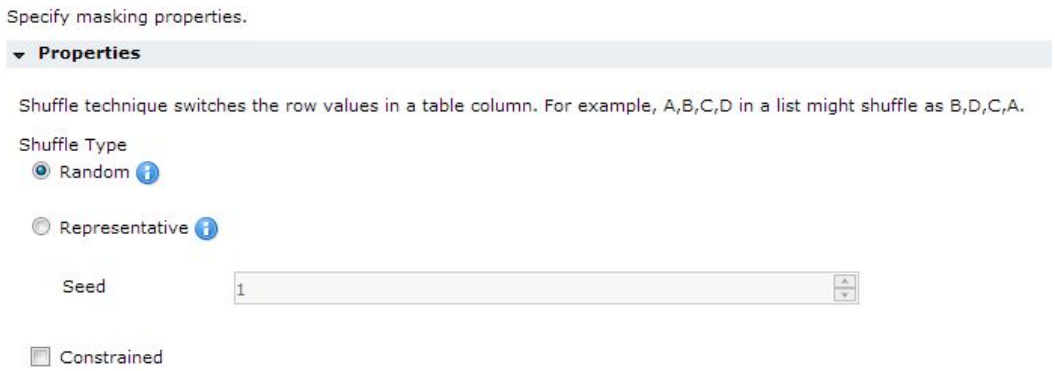
You cannot use shuffle masking when both the source and the target use Hadoop HDFS connections.

Note: If the source file might have empty strings in the shuffle column, set the **Null and Empty Spaces** option to Treat as Value in the rule exception handling. When you set the option to Treat as Value, the PowerCenter Integration Service masks the space or the null value with a valid value. The default is to skip masking the empty column.

Shuffle Masking Parameters

You can configure masking parameters to determine if shuffle masking is repeatable, the masking is repeatable for one workflow run, or the masking is random. You can also configure a lookup to ensure that replacement values originate from rows that contain specific values.

The following image shows Data Masking parameters that appear when you configure a Shuffle data masking rule:



The following table describes the parameters that you can configure for shuffle masking:

Parameter	Description
Shuffle Type	Select random or representative shuffling: - Random. Shuffle values from one row to another without checking if the target values are unique for each source value. For example, the Integration Service masks 12345 with 65432 in a row. The Integration Service can also replace 33333 with 12345 in another row. - Representative. All source rows with the same value receive the same shuffle value. When the Integration Service replaces 12345 with 65432, then it can use 65432 as a mask value for any row with a 12345 source value. Representative masking does not save values between workflow runs. Use repeatable masking to return the same values between workflow runs.
Seed	Starting point for creating repeatable output. Enter a number between 1 and 999. Default is 1. Enabled when Representative Shuffle Type is selected.
Constrained	Restricts applying shuffle masking to rows that are constrained by another column. For example, shuffle employee names based on gender. Or, shuffle addresses within the same city. Choose the constraint column when you assign the rule to columns in a project.

SIN Masking

You can mask a Social Insurance number (SIN) that is nine digits. The digits can be delimited by any set of characters. Mask string datatypes with social insurance masking.

If the number contains no delimiters, the masked number contains no delimiters. Otherwise the masked number has the following format: `xxx-xxx-xxx`.

You can define the first digit of the masked SIN.

Enable **Start Digit** and enter the digit. The PowerCenter Integration Service creates masked Social Insurance numbers that start with the number that you enter.

You can configure repeatable masking for Social Insurance numbers. To configure repeatable masking for SIN numbers, click Repeatable Output and enter a Seed Value.

SSN Masking

SSN masking applies a built-in mask format to change Social Security numbers. Mask string datatypes with SSN masking.

The PowerCenter Integration Service generates valid Social Security numbers. To avoid generating numbers that the Social Security Administration has already issued, you can download the latest version of the High Group List at the following location:

```
http://www.ssa.gov/employer/highgroup.txt
```

The PowerCenter Integration Service accesses the latest High Group List from the following location:

```
<PowerCenter Installation Directory>\infa_shared\SrcFiles\highgroup.txt
```

The PowerCenter Integration Service generates Social Security numbers that are not on the High Group List.

The SSN masking accepts any SSN format that contains nine digits. You can delimit the digits with any set of characters. For example, the SSN masking accepts the following format:

```
+=54-*9944$#789-,*()”
```

You can configure repeatable masking for Social Security numbers. Select the **Repeatable Output** option, select the **Seed** option, and enter a value.

The PowerCenter Integration Service returns deterministic Social Security numbers with repeatable masking. The PowerCenter Integration Service cannot return all unique Social Security numbers because it cannot return valid Social Security numbers that the Social Security Administration has issued.

SSN Randomization

The Social Security Administration developed a method to randomize the nine-digit SSN. This method eliminates the geographical significance of the first three digits of SSN and protects the integrity of the Social Security numbers.

When you configure SSN masking parameters, you can use the SSN randomization technique to mask a Social Security number.

Substitution Masking

Substitution masking replaces a column of data with similar but unrelated data from a dictionary. Mask date, numeric, and string data types with substitution masking.

Use substitution masking to mask string data with realistic output. For example, if you want to mask address data, you specify a dictionary file that contains addresses. If you want to mask a Social Security number, you can specify the InvalidSSN dictionary file that contains Social Security numbers that are not valid.

Substitution is an effective way to replace production data with realistic test data. When you configure substitution masking, select the relational or flat file dictionary that contains the substitute values. The PowerCenter Integration Service performs a lookup on the dictionary and replaces source data with data from the dictionary.

When you assign a substitution masking rule to a column, you can specify the rule assignment parameters.

The following table describes the rule assignment parameters that you can configure:

Parameter	Description
Lookup Condition	The column name in the source table you can refer to match with the column in the dictionary. This field is optional.
Unique Substitution Column	The column name in the source table to substitute with unique data. This field is optional.

You can substitute data with repeatable or non-repeatable values. When you choose repeatable values, the PowerCenter Integration Service produces deterministic results for the same source data and seed value. You must configure a seed value to substitute data with deterministic results. The PowerCenter Integration Service maintains a storage table of source and masked values for repeatable masking. You can specify the storage table you want to use when you generate a workflow.

Substitution Masking Parameters

You can substitute data with repeatable or non-repeatable values.

When you choose repeatable values, the PowerCenter Integration Service produces deterministic results for the same source data and seed value. You must configure a seed value to substitute data with deterministic results.

You can configure the following substitution masking parameters:

Parameter	Description
Repeatable Output	Returns deterministic results between sessions. The PowerCenter Integration Service saves masked values in the storage table.
Seed	A start number that the PowerCenter Integration Service uses to return deterministic data.
Unique Substitution Data	Replaces the target column with unique masked values for every unique source column value. If there are more values in the source than in the dictionary file, the data masking operation fails because the dictionary file does not contain sufficient unique values to substitute the data. For security, the default is nonunique substitution.
Optimize Dictionary Usage	Required if you select the Repeatable Output option. Increases the usage of masked values from the dictionary. This option is not available if you enable the Unique Substitution Data option.
Dictionary Information	Required. Configuration of the flat file or relational table that contains the substitute data values. Configure the following parameters: - Dictionary. Displays the flat file or relational table name that you select. - Masked Value. The column returned to the masking rule. - Lookup Column. The source data column to use in the lookup. - Serial Number Column. The column in the dictionary that contains the serial number.

URL Masking

URL masking applies a built-in mask format to change URL data. Mask string datatypes with URL masking.

The PowerCenter Integration Service parses a URL by searching for the `://` string and parsing the substring to the right of it. The source URL must contain the `://` string. The source URL can contain numbers and alphabetic characters.

The PowerCenter Integration Service does not mask the protocol of the URL. For example, if the URL is `http://www.yahoo.com`, the PowerCenter Integration Service can return `http://MgL.aHjCa.VsD/`. The PowerCenter Integration Service might generate a URL that is not valid.

Note: The PowerCenter Integration Service always returns ASCII characters for a URL.

Name Substitution Example

You want to mask employee names and you want to preserve the gender and nationality of the names in the masked data.

You create substitution masking rules to mask first names based on gender and nationality. Substitution masking replaces a column of data with similar but unrelated data from a dictionary. You use substitution masking to mask string data with realistic output. You use advanced masking to apply multiple masking techniques to a column.

Complete the following steps to create rules to mask names based on gender and nationality:

1. Add a dictionary in Test Data Manager.
2. Create a substitution rule that substitutes first names based on gender.
3. Create an advanced masking rule to substitute first names based on nationality.

Add a Dictionary in Test Data Manager

Add a dictionary in Test Data Manager to use for substitution masking.

The dictionary must contain country, gender, and first name columns. You can use a flat file or a relational dictionary. Test Data Manager uses the dictionary to substitute data.

The following text is a sample from the flat file dictionary that you use to mask the employee names:

```
SNO, COUNTRY, GENDER, FIRSTNAME
1, US, M, Adam
2, US, F, Sarah
3, JP, M, Mahito
4, JP, F, Kyoko
```

Adding a Relational Dictionary

When you add a relational dictionary, you define the connection to the dictionary.

1. In the **Administrator | Dictionaries** view, click **Actions** > **New Dictionary**.

   The **New Dictionary** tab appears.
2. Enter the name of the dictionary, an optional description of the dictionary, and the type of the dictionary.
3. Click **Select** to define a connection.

   The **Select Relational Dictionary** dialog box appears.
4. Select a datasource connection from the menu, and click **Next**.
5. Select a datasource, and click **Next**.
6. Select a table from the list of tables in the datasource, and click **Finish**.
7. Review the **Connection**, **Schema**, and **Table** properties you selected.
8. Click **Save**.

   A tab with the dictionary properties opens and the dictionary appears in the **Administrator | Dictionaries** view.

Adding a Flat File Dictionary

When you add a flat file dictionary, you can select a dictionary from the local system.

1. In the **Administrator | Dictionaries** view, click **Actions** > **New Dictionary**.

   The **New Dictionary** tab appears.
2. Enter a name and optional description for the dictionary. Select **Flat File** as the dictionary type.
3. Click **Choose File** to browse to the location of the dictionary. Select the dictionary file, and click **Open**.
4. Optionally, select to overwrite the file if a dictionary exists in the `<TDM installation directory>/TDM/infa_shared/LkpFiles` directory.
5. Select the file encoding type.
6. Click **Save**.

   A tab with the dictionary properties opens and the dictionary appears in the **Administrator | Dictionaries** view.

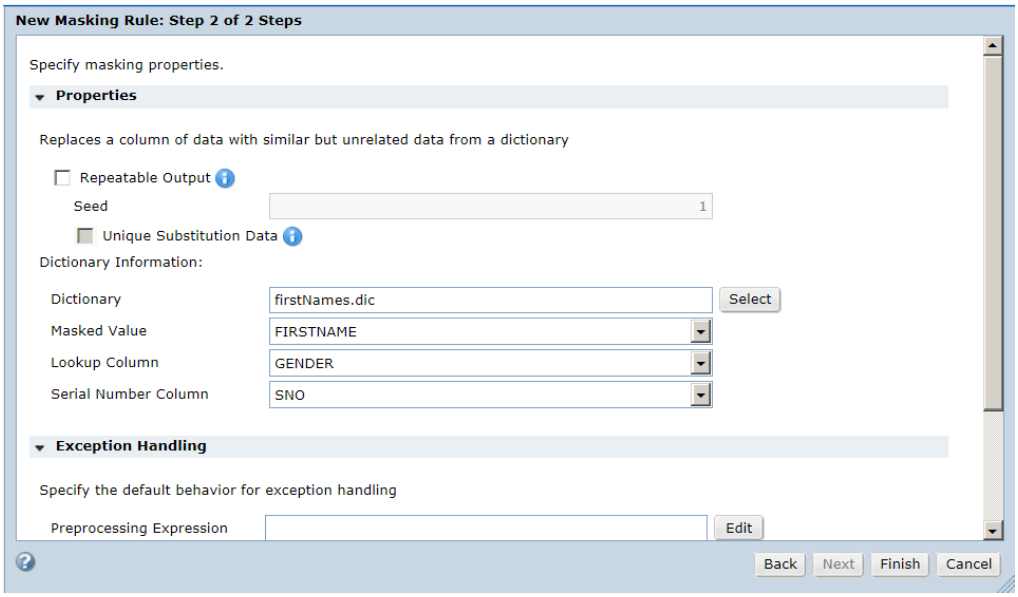
Creating the Substitution Rule

Create a substitution rule that substitutes first name based on gender.

1. In the **Policies** view, click **Actions** > **New** > **Masking Rule**.

   The **New Rule** window appears.
2. Enter a name and optional description for the rule.
3. Select the string datatype and standard substitution masking type. Select override allowed.
4. Click **Next**.
5. Enter the following substitution masking parameters:
   - Dictionary. Select the dictionary to use for the masking rule.
   - Masked Value. Select FIRSTNAME.
   - Lookup column. Select GENDER.
   - Serial number column. Select the serial number column.

   The following image shows the substitution masking parameters:



6. Click **Finish**.

   The rule appears in the **Policies** view.

Creating the Advanced Masking Rule

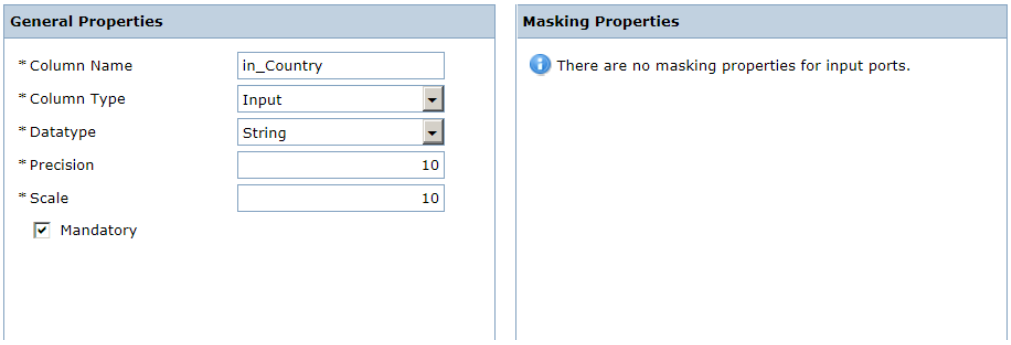
Create an advanced rule that substitutes first names based on nationality with a lookup condition on gender.

1. In the **Policies** view, click **Actions** > **New** > **Masking Rule**.

   The **New Masking Rule** window appears.
2. Enter a name and optional description for the rule.
3. Select the advanced masking type.
4. Click **Next**.
5. Click **Create Input Column** in the **Specify Masking Properties** window.

   The **Add Column** window appears.
6. Enter the following general properties:
   - Column name. Enter in_Country.
   - Column type. Select input.
   - Datatype. Select string.
   - Precision. Select 10.
   - Scale. Select 10.
   - Mandatory. Select the check box.

   The following image shows the in_Country column properties:



7. Click **OK**.

   The in_Country input column appears in the list of input columns.
8. Click **Create Input Column**.

   The **Add Column** window appears.
9. Enter the following general properties:
   - Column name. Enter in_FirstName.
   - Column type. Select input.
   - Datatype. Select string.
   - Precision. Select 10.
   - Scale. Select 10.

- Mandatory. Select the check box.

10. Click **OK**.

    The in_FirstName input column appears in the list of input columns.

11. Click **Create Input Column**.

    The **Add Column** window appears.

12. Enter the following general properties:
    - Column name. Enter in_Gender.
    - Column type. Select input.
    - Datatype. Select string.
    - Precision. Select 10.
    - Scale. Select 10.
    - Mandatory. Select the check box.

13. Click **OK**.

    The in_Gender input column appears in the list of input columns.

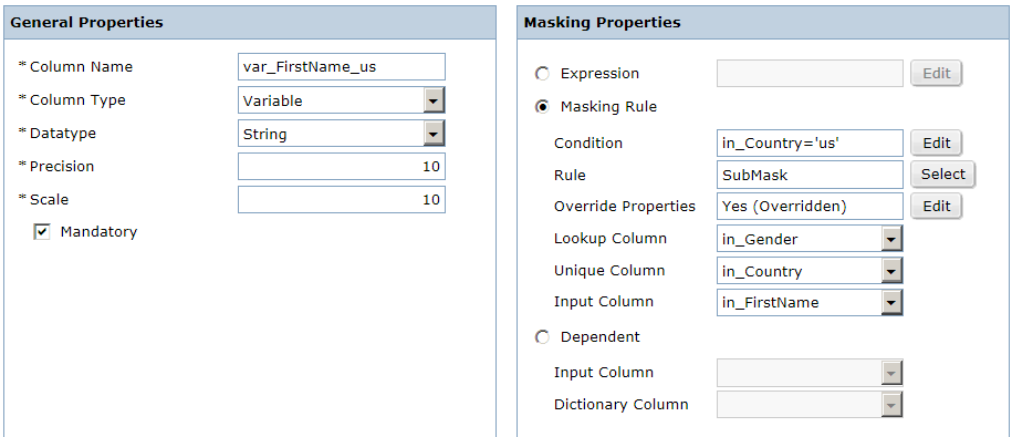
14. Click **Create Input Column**.

    The **Add Column** window appears.

15. Enter the following general properties:
    - Column name. Enter var_FirstName_us.
    - Column type. Select variable.
    - Datatype. Select string.
    - Precision. Select 10.
    - Scale. Select 10.
    - Mandatory. Select the check box.

16. Select the masking rule masking property. Configure the following masking properties:
    - Condition. Enter in_Country='us'.
    - Rule. Select the substitution rule you created in Step 2.
    - Override Properties. Click **Edit**. Enable **Repeatable Output** and **Unique Substitution Data** options, and click **Save**. The property appears as Yes (Overridden).
    - Lookup column. Select in_Gender.
    - Unique column. Selct in_Country.
    - Input column. Select in_FirstName.

The following image shows the var_FirstName_us column properties:



17. Click **OK**.

    The var_FirstName_us variable column appears in the list of variable columns.

18. Click **Create Input Column**.

    The **Add Column** window appears.

19. Enter the following general properties:
    - Column name. Enter var_FirstName_jp.
    - Column type. Select variable.
    - Datatype. Select string.
    - Precision. Select 10.
    - Scale. Select 10.
    - Mandatory. Select the check box.

20. Select the masking rule masking property. Configure the following masking properties:
    - Condition. Enter in_Country='jp'.
    - Rule. Select the substitution rule you created in Step 2.
    - Override Properties. Click **Edit**. Enable **Repeatable Output** and **Unique Substitution Data** options, and click **Save**. The property appears as Yes (Overridden).
    - Lookup column. Select in_Gender.
    - Unique column. Selct in_Country.
    - Input column. Select in_FirstName.

21. Click **OK**.

    The var_FirstName_jp variable column appears in the list of variable columns.

22. Click **Create Input Column**.

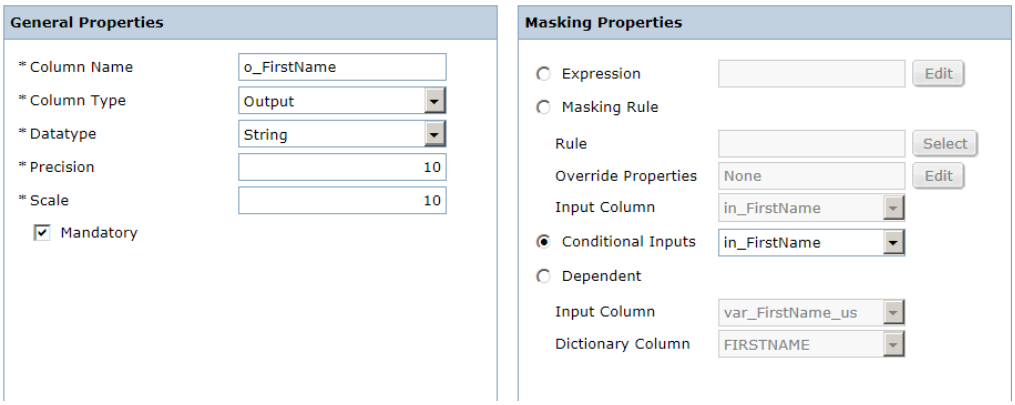
    The **Add Column** window appears.

23. Enter the following general properties:
    - Column name. Enter o_FirstName.
    - Column type. Select output.
    - Datatype. Select string.

- Precision. Select 10.
- Scale. Select 10.
- Mandatory. Select the check box.

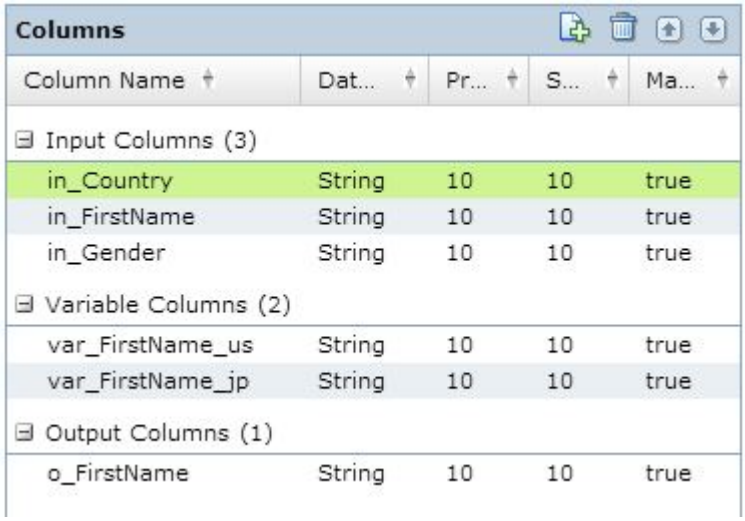
24. Select the conditional inputs masking property. Configure the following masking property:
    - Conditional input. Select in_FirstName.

    The following image shows the o_FirstNames column properties:



25. Click **OK**.

    The o_FirstName variable column appears in the list of output columns.
    The following image shows the rule columns:



Columns

Column Name	Dat...	Pr...	S...	Ma...
Input Columns (3)				
in_Country	String	10	10	true
in_FirstName	String	10	10	true
in_Gender	String	10	10	true
Variable Columns (2)				
var_FirstName_us	String	10	10	true
var_FirstName_jp	String	10	10	true
Output Columns (1)				
o_FirstName	String	10	10	true

26. Click **Next**.

    Review the rule map that appears.

27. Click **Finish**.

    The rule appears in the **Policies** view.

Shuffle Address Example

You want to mask employee addresses and keep the ZIP code unmasked.

Create a shuffle rule and an advanced masking rule to shuffle addresses and leave the ZIP code unmasked. Shuffle masking masks the data in a column with data from the same column in another row of the table. Shuffle masking switches all the values for a column in a file or database table. You can restrict which values to shuffle based on a lookup condition or a constraint.

Create shuffle masking rules with a dictionary that contains three address columns and a ZIP code column. The ZIP code remains unmasked. The three address columns shuffle, but remain consistent.

The following text shows a sample flat file with the required columns:

```
SNO,AddressLine1,AddressLine2,AddressLine3,ZIP
1,3290 Apple Lane,Chillicothe,IL,61523
2,7760 Ash Street,Dallas,TX,75240
3,2229 Ash Street,Moscow,TN,38057
4,6698 Caldwell Road,Rochester,NY,14620
```

Complete the following steps to create rules to shuffle addresses with the ZIP code unmasked in Test Data Manager:

1. Create a shuffle rule.
2. Create an advanced masking rule that keeps the ZIP code column unmasked and shuffles the address columns together.

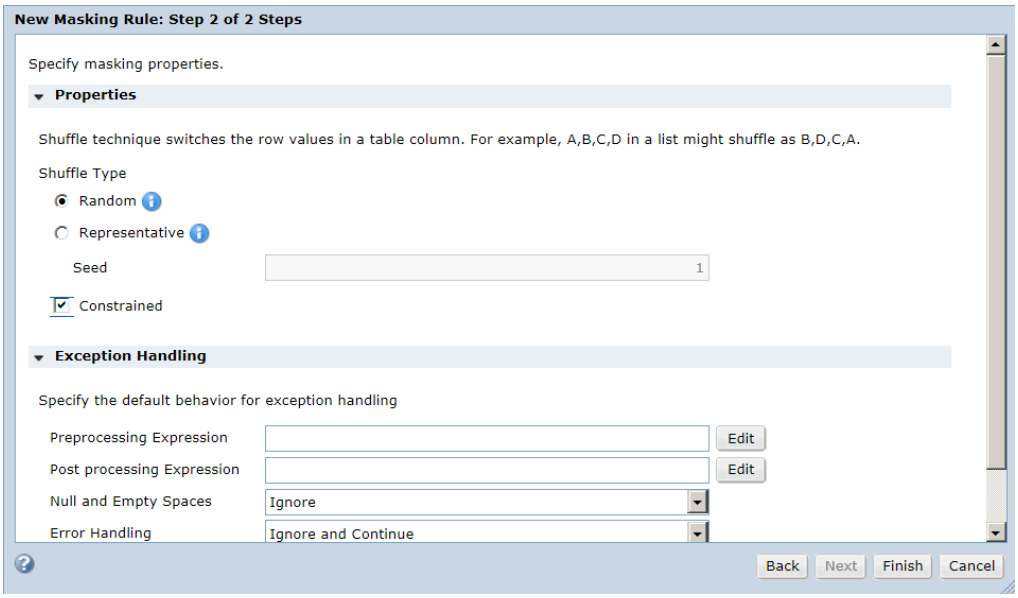
Creating the Shuffle Rule

Create a shuffle rule in Test Data Manager.

1. In the **Policies** view, click **Actions** > **New** > **Masking Rule**.

   The **New Rule** window appears.
2. Enter a name and optional description for the rule.
3. Select the string datatype and standard shuffle masking type. Select override allowed.
4. Click **Next**.
5. Enter the following shuffle masking parameters:
   - Shuffle type. Select Random.
   - Constrained. Select Constrained.

The following image shows the shuffle masking parameters:



6. Click **Finish**.

   The rule appears in the **Policies** view.

Create the Advanced Masking Rule

Create an advanced masking rule that shuffles address lines.

The advanced rule shuffles three address line columns together and keeps the ZIP code column unmasked. The rule uses shuffle masking with a lookup on AddressLine3 and dependent masking on AddressLine1 and AddressLine2.

In the rule editor, you create input columns, variable columns, and output columns.

Setting Up the Advanced Masking Rule

Set up the advanced masking rule.

1. In the **Policies** view, click **Actions** > **New** > **Masking Rule**.

   The **New Masking Rule** window appears.
2. Enter a name and optional description for the rule.
3. Select the advanced masking type.
4. Click **Next**.

Creating the Input Columns

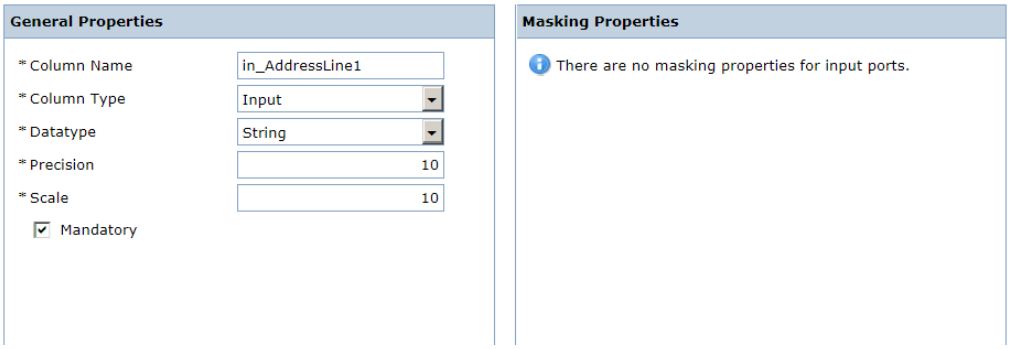
Create input columns in the advanced masking rule.

1. Create an input column. Click **Create Input Column** in the **Specify Masking Properties** window.

   The **Add Column** window appears.

2. Enter the following general properties:
   - Column name. Enter i_AddressLine1.
   - Column type. Select Input.
   - Datatype. Select String.
   - Precision. Select 10.
   - Scale. Select 10.
   - Mandatory. Select the check box.

   The following image shows the i_AddressLine1 column properties:



3. Click **OK**.

   The i_AddressLine1 input column appears in the list of input columns.
4. Create an input column with the following general properties:
   - Column name. Enter i_AddressLine2.
   - Column type. Select Input.
   - Datatype. Select String.
   - Precision. Select 10.
   - Scale. Select 10.
   - Mandatory. Select the check box.
5. Create an input column with the following general properties:
   - Column name. Enter i_AddressLine3.
   - Column type. Select Input.
   - Datatype. Select String.
   - Precision. Select 10.
   - Scale. Select 10.
   - Mandatory. Select the check box.
6. Create an input column with the following general properties:
   - Column name. Enter i_ZIP.
   - Column type. Select Input.
   - Datatype. Select String.
   - Precision. Select 10.
   - Scale. Select 10.

- Mandatory. Select the check box.

Creating the Variable Columns

Create variable columns in the advanced masking rule.

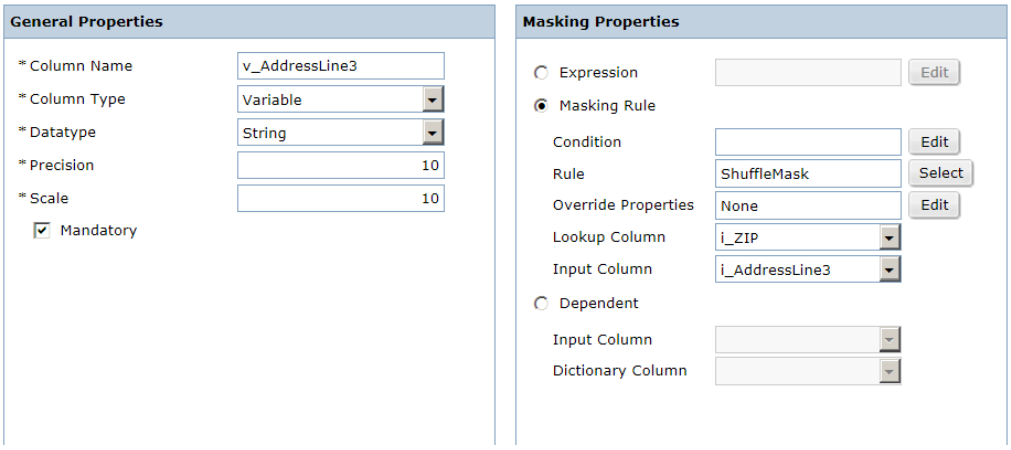
1. Create a variable column to shuffle AddressLine3 based on ZIP. Click **Create Input Column**.

   The **Add Column** window appears.

2. Enter the following general properties:
   - Column name. Enter v_AddressLine3.
   - Column type. Select Variable.
   - Datatype. Select String.
   - Precision. Select 10.
   - Scale. Select 10.
   - Mandatory. Select the check box.

3. Select the masking rule masking property. Configure the following masking properties:
   - Condition. Leave blank.
   - Rule. Select the shuffle rule that you created in Step 2.
   - Override Properties. Select None.
   - Lookup column. Select i_ZIP.
   - Input column. Select i_AddressLine3.

   The following image shows the v_AddressLine3 column properties:



4. Click **OK**.

   The v_AddressLine3 variable column appears in the list of variable columns.

5. Create a variable column to mask AddressLine2 based on the masked value of AddressLine3. Click **Create Input Column**.

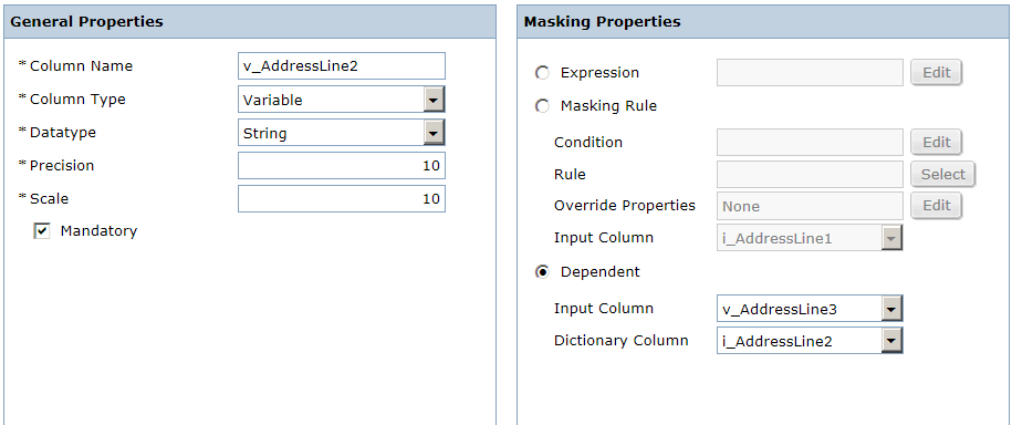
   The **Add Column** window appears.

6. Enter the following general properties:
   - Column name. Enter v_AddressLine2.
   - Column type. Select Variable.

- Datatype. Select String.
- Precision. Select 10.
- Scale. Select 10.
- Mandatory. Select the check box.

7. Select the dependent masking property. Configure the following masking properties:
   - Input column. Select v_AddressLine3.
   - Dictionary column. Select i_AddressLine2.

   The following image shows the v_AddressLine2 column properties:



8. Click **OK**.

   The v_AddressLine2 variable column appears in the list of variable columns.

9. Create a variable column to mask AddressLine1 based on the masked value of AddressLine3. Click **Create Input Column**.

   The **Add Column** window appears.

10. Enter the following general properties:
    - Column name. Enter v_AddressLine1.
    - Column type. Select Variable.
    - Datatype. Select String.
    - Precision. Select 10.
    - Scale. Select 10.
    - Mandatory. Select the check box.

11. Select the dependent masking property. Configure the following masking properties:
    - Input column. Select v_AddressLine3.
    - Dictionary column. Select i_AddressLine1.

12. Click **OK**.

    The v_AddressLine1 variable column appears in the list of variable columns.

Creating the Output Columns

Create output columns in the advanced masking rule.

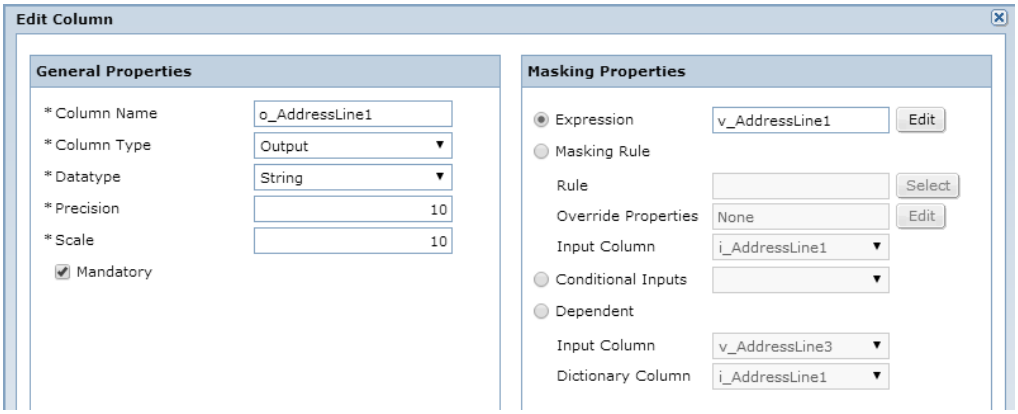
1. Click **Create Input Column**.

   The **Add Column** window appears.

2. Enter the following general properties:
   - Column name. Enter o_AddressLine1.
   - Column type. Select Output.
   - Datatype. Select String.
   - Precision. Select 10.
   - Scale. Select 10.
   - Mandatory. Select the check box.

3. Select the dependent masking property. Configure the following dependent properties:
   - Input column. Select v_AddressLine3.
   - Dictionary column. Select i_AddressLine1.

   The following image shows the o_AddressLine1 column properties:



4. Click **OK**.

   The o_AddressLine1 variable column appears in the list of variable columns.

5. Click **Create Input Column**.

   The **Add Column** window appears.

6. Enter the following general properties:
   - Column name. Enter o_AddressLine2.
   - Column type. Select Output.
   - Datatype. Select String.
   - Precision. Select 10.
   - Scale. Select 10.
   - Mandatory. Select the check box.

7. Select the dependent masking property. Configure the following dependent properties:
   - Input column. Select v_AddressLine3.
   - Dictionary column. Select i_AddressLine2.

8. Click **OK**.

   The o_AddressLine2 variable column appears in the list of variable columns.

9. Click **Create Input Column**.

   The **Add Column** window appears.

10. Enter the following general properties:

    - Column name. Enter o_AddressLine3.
    - Column type. Select Output.
    - Datatype. Select String.
    - Precision. Select 10.
    - Scale. Select 10.
    - Mandatory. Select the check box.

11. Select the dependent masking property. Configure the following dependent properties:

    - Input column. Select v_AddressLine3.
    - Dictionary column. Select i_AddressLine3.

12. Click **OK**.

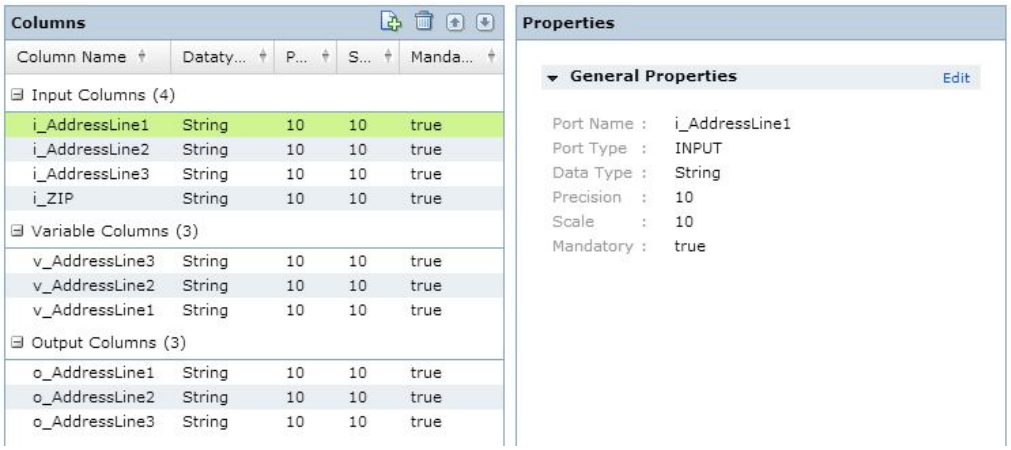
    The o_AddressLine3 variable column appears in the list of variable columns.

Saving the Advanced Masking Rule

Save the advanced masking rule.

1. Verify that all the rule columns that you created are visible in the columns list.

   The following image shows the rule columns:



2. Click **Finish**.

   The rule appears in the **Policies** view.

CHAPTER 9

Data Generation

This chapter includes the following topics:

- Data Generation Overview, 181
- Data Generation Components, 182
- Data Generation Task Flow, 183
- Data Generation Rule Types, 183
- Default Settings, 184
- Define Default Generation Rules, 185
- Standard Generation Rules, 185
- Custom Generation Rules, 187
- Ad Hoc Generation Rules, 188
- Advanced Generation Rules, 190
- Conditional Constraints, 192
- Data Generation Rule Assignments, 193
- Data Generation for XSD Sources, 197
- Data Generation Plans and Workflows, 198

Data Generation Overview

Use data generation to create realistic test data for test environments. You can define generation rules that define the logic to generate data.

Import metadata into a project to define the type of data that you want to generate. You assign generation rules that you create or default rules to the target columns to generate data based on the data type of the column. When you create a rule, you can choose a generation technique and configure parameters to create random test data.

If a table name or a column name contains special characters, the data generation workflow fails.

To implement data generation, you create a data generation plan and a workflow from the plan. If the target is a flat file, you can configure test tool integration properties in the plan. Configure test tool integration to copy the results to a location in an integrated HP ALM server. To store the test data along with the metadata in the test data warehouse, select the test data warehouse as the target connection.

Data Generation Example

You work for an organization that sells plane tickets. You want to generate data in tables that contain customer information such as identification number, status of membership, and address. You want additional tables to store ticket details such as ticket number and flight number. To generate the data, you can perform the following tasks:

- Create data generation rules that load dictionary values such as names into the tables.
- Create random number strings for ticket numbers.
- Create a numeric sequence for identification numbers.
- Use a reference lookup for values such as airport codes.
- Create projects to import metadata, enable relationships, and create entities.
- Make rule assignments, create a plan, and run the plan to generate the data.

Data Generation Components

To perform data generation operations, assign rules to columns. Import metadata to define the columns in the target database.

The following table describes the components that you create to implement data generation:

Component	Description
Assignments	The allocation of rules to a column to generate the column data.
Plan	Defines data generation operations. You import the target schema metadata into the repository in the plan and configure the settings.
Rule	Defines the data generation technique and parameters. A generation technique defines the logic to generate the data. Generation parameters define how a generation technique in a rule generates data. You can set an override option in a rule that defines whether users can modify the generation parameters for the rule when they assign the rule to the columns in a target.
Table	The target table in which you assign a generation rule.
Entity	Defines a set of tables that are related based on physical or logical constraints. You cannot perform data generation operations that include an entity if the primary key column is a computed column.

Data Generation Task Flow

To implement data generation operations, assign generation rules to target columns. Create a plan and add rules to the plan. Generate a workflow from the plan and run the workflow to generate data in the target database.

To create the components that you need in a data generation plan, perform the following high-level steps:

1. Create generation rules that define the generation techniques and configure the parameters. Define default rules that generate data for columns that you do not assign a generation rule to.
2. Create a project and import metadata into the project.
3. Optional. Create parameters to use in the plan.
4. If you want to restrict the generated values in a column based on values in another column, create a conditional constraint.
5. Add generation rules to the project and assign the rules to columns in the project.
6. Create a plan and add data generation components to the plan.
7. Generate and run the plan. View the progress of the workflow in the **Monitor** view.

Data Generation Rule Types

A data generation rule is a rule that defines how you want to generate data in the target table. When you create a data generation rule, you choose a generation technique and configure generation parameters.

You can create the following data generation rule types:

Standard

A data generation rule that uses predefined generation techniques.

Custom

A data generation rule that generates data based on PowerCenter mapplets logic.

Advanced

A data generation rule that uses a combination of generation techniques that generates data in target columns based on values of more than one input column.

Ad Hoc

A data generation rule that you create within a project for a column in the target table.

Standard rules, custom rules. and advanced rules are global rules. Global rules are rules that you create in the **Policies** view of Test Data Manager. When you create a global generation rule in the **Policies** view, you specify a rule name. This rule is available for projects that you create. You assign the rule to one or more columns in the project that you want to generate data into.

Ad hoc generation rules are rules that you create in the **Define | Data Generation** view of a project. Ad hoc rules do not have a name, and you cannot move an ad hoc rule. You can create ad hoc generation rules within the projects. You select the target columns for an ad hoc generation rule when you create the rule. You can select a standard generation technique for an ad hoc generation rule or you can select an ad hoc generation technique. You can use reference lookup, effective dates, and expression generation techniques only in an ad hoc generation rule. In an ad hoc rule, if you want to generate numbers in a column that uses string data type, you can change the data type to numeric.

After you install TDM and log in to Test Data Manager, TDM configures the default generation rules that comes with the installation package.

A data generation technique defines the logic that generates the data. When you create a generation rule, the techniques vary based on datatype you select. Generation parameters are options that you configure for a generation technique. For example, you can define different start dates and end dates for random date generation.

You can enable users to override generation parameters for a rule. For example, you create a rule that generates random dates from 2011 to 2012. You set the override option for the rule. When a user assign this rule to a target column, the user can select Saturday and Sunday as exclude dates so that the rule generates random weekday dates from 2011 to 2012.

Default Settings

Use default settings to generate test data for the columns that you have not assigned specific generation rules to.

When you create a project and import target metadata, the project contains the default generation rules for all data types that a TDM administrator configures. When you open a project, you can view the default data generation rules, default primary key generation rules, and default unique key generation rules for all data types from the **Overview | Properties** tab. Edit the default settings if you want to change the default generation rules.

The following table describes the properties to configure the default settings:

Property	Description
Default Rule for Numeric Datatype	The default rule that you can select to generate data for numeric data types.
Default Rule for String Datatype	The default rule that you can select to generate data for string data types.
Default Rule for Date Datatype	The default rule that you can select to generate data for date data types.
Default Rule for Binary Datatype	The default rule that you can select to generate data for binary data types.
Default Primary Key Rule for Numeric Datatype	The default primary key rule that you can select to generate data for numeric data types.
Default Primary Key Rule for String Datatype	The default primary key rule that you can select to generate data for string data types.
Default Primary Key Rule for Date Datatype	The default primary key rule that you can select to generate data for date data types.
Default Primary Key Rule for Binary Datatype	The default primary key rule that you can select to generate data for binary data types.
Default Unique Key Rule for Numeric Datatype	The default unique key rule that you can select to generate data for numeric data types.
Default Unique Key Rule for String Datatype	The default unique key rule that you can select to generate data for string data types.

Property	Description
Default Unique Key Rule for Date Datatype	The default unique key rule that you can select to generate data for date data types.
Default Unique Key Rule for Binary Datatype	The default unique key rule that you can select to generate data for binary data types.
Apply Default Rule to	The default rule that you can apply to all the columns without generation rules. You can also apply the default rule to the columns that do not have null values.
Default Character Set	The character set that you can select. Default is ASCII.
Minimum	The minimum number of child records for a parent record. Default is 1.
Maximum	The maximum number of child record for a parent record. Default is 1.
Parent Records Without Children	The number of parent records that have no child.
Child Records Without a Parent	The number of child records that have no parent.

Define Default Generation Rules

To generate test data, you can apply auto rule, default data domain rule, default global rule, default primary key rule, or default unique key rule to a target column.

View the default generation rules from the **Overview** | **Properties** tab. Edit the default settings if you want to change the default generation rules that you want to assign to the target columns.

You can apply the default generation rules from the **Define** | **Data Generation** page. The default data domain rule is the default rule that you configure when you create and run a data domain profile.

After you run a column profile, you can review and finalize the profile results. You can assign auto rule to columns that contain Yes in the Inferred Rule column. When you apply auto rule to a target column, TDM populates the target column with the approved profile results. You must not apply an auto rule to a primary key column because a primary key column cannot contain null values.

Standard Generation Rules

A standard generation rule is a global rule that applies a system-defined generation technique. A standard generation rule has one output column.

When you create a standard generation rule, you select the generation technique. You can define one generation technique in a standard rule and you can apply the rule to one column at a time. You can apply the rule multiple times to different columns.

You can use standard generation techniques to generate string, numeric, date, and binary datatypes. The techniques vary based on the datatype.

For example, you create a standard generation rule with the random date technique to generate birth dates. You can assign the rule multiple times in a project to generate birth dates in more than one column or you can use the rule in multiple projects.

Creating a Standard Generation Rule

To create a standard generation rule, you choose a generation technique based on the data type and configure the generation parameters to apply the technique.

1. Click **Policies** to access the **Policies** view.
2. Click **Actions** > **New** > **Generation Rule**.

   The **Rule Wizard** appears.
3. Enter a name and optional description for the rule.
4. Select the data type of the column to apply the generation rule to.
5. Select the standard generation rule.
6. To enable users to override generation parameters for a rule, select **Override Allowed**.
7. Click **Next**.
8. Enter the generation parameters.

   Note: The parameters that appear depend on the **Generation Technique** that you select.
9. Enter the exception test data options. Configure whether to allow null or empty spaces. Configure whether to generate data that is not valid.
10. Click **Finish**.

Editing a Generation Rule

You can edit a global generation rule to change the rule parameters.

1. Click **Policies**.
2. Click the name of the generation rule that you want to edit.

   The rule opens in a tab.
3. Click **Edit**.
4. Edit the parameters that you want to change.
5. To save the rule, click **OK**.

   If the generation rule has assignments, the **Impacted Objects** dialog box appears with the list of affected columns, plans, and data domains.
6. To download the list of affected objects, click **Export**, and save the .csv file.
7. To save the changes, click **Continue**.

   To update the changes in a plan, generate and run the plan again.

Overriding a Generation Rule

After you assign a generation rule to a column, you can override the rule properties for the column. You can override rule properties for global generation rules.

1. Open a project and click **Define** > **Data Generation**.
2. Select the column that you need to override the rule properties for.

3. Click **Actions** > **Override**.

   The **Generation Rule Override** window appears.
4. Update the values in the generation rule properties.
5. Click **Save** for each column that you update.

   The **Override** column value shows Yes-Overridden for the column that you updated.

Deleting a Generation Rule

You can delete a global generation rule. If the rule has assignments, you must make changes to the plan, and generate and run the plan again.

1. Click **Policies**.
2. Click the box to the left of the generation rule that you want to delete.
3. Click **Actions** > **Delete**.

   If the generation rule has no assignments, the **Delete Generation Rule** dialog box appears. If the generation rule has assignments, the **Impacted Objects** dialog box appears with the list of affected plans, columns, and data domains.
4. To delete the generation rule that has no assignments, click **OK**.
5. To delete the generation rule that contains assignments, click **Continue**. To download the list of affected objects, click **Export**, and save the .csv file.

   To update the changes in a plan, generate and run the plan again.

Custom Generation Rules

A custom generation rule is a global rule that generates data based on PowerCenter mapplet logic.

You can import a mapplet to create a custom generation rule. You must mark at least one input and one output column as required. Use custom rules to generate data for string, numeric, and date data types.

For example, an EMPLOYEES table contains columns for first name and last name. You create a custom generation rule to combine the first and last name of an employee to generate the full name.

You can also create custom rule with mapplets when you want to use PowerCenter transformations, such as an SQL transformation or a Stored Procedure transformation. You can also use custom mapplets when you need to use the logic to generate records or generate output with the user defined procedure.

Creating a Custom Generation Rule

Create a custom rule using PowerCenter mapplet to generate data.

1. To access the **Policies** view, click **Policies**.
2. Click **Actions** > **New** > **Generation Rule**.

   The **Rule Wizard** appears.
3. Enter a name and optional description for the rule.
4. Select the **Custom Rule** generation type.
5. To select a mapplet from the local folder, click **Browse**.

6. Click **Next**.

   The **Configure Mandatory Columns** dialog box appears with a list of input columns and output columns that are available in the mapplet.

7. To mark the columns as required fields, select the columns, and click **Set Mandatory**.

   You must at least select one input column and one output column as required fields.

8. Click **Finish**.

   The custom rule appears as another tab. Review and close the tab.

Ad Hoc Generation Rules

An ad hoc generation rule is a data generation rule that you create within a project. You can create an ad hoc generation rule for a column in the target table.

An ad hoc rule does not have a name. You cannot move an ad hoc rule because you use target columns to define the rule.

An ad hoc generation rule includes the standard data generation techniques and the techniques such as reference lookup generation, effective dates generation, and expression generation. An ad hoc rule can have more than one target column, based on the generation technique that you select.

When you create an ad hoc rule, the generation techniques that you can select depend on the data type of the column. To create an ad hoc rule, you can use a standard or an ad hoc generation technique. The ad hoc generation techniques include expression generation, reference lookup generation, and effective dates generation.

For example, an Airport table contains airport codes. You create a reference lookup rule on a From_City column that generates data from the Airport table and inserts it into the From_City column in the target table.

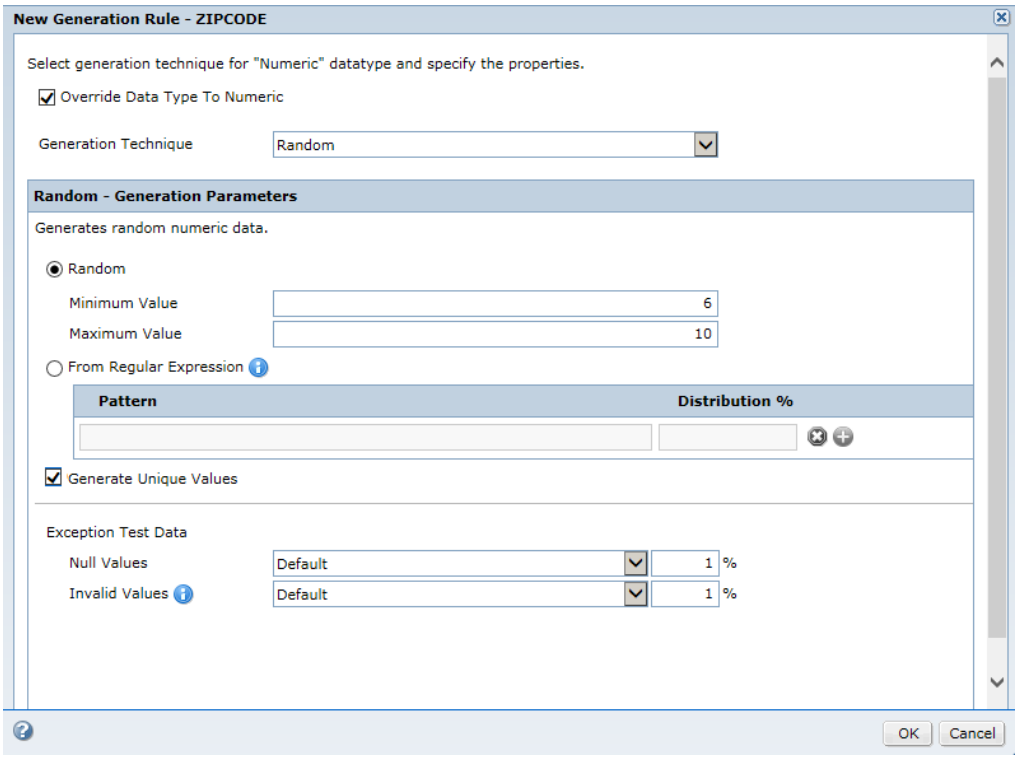
In an ad hoc rule, if you want to generate numbers in a column that uses string data type, you can change the data type to numeric. For example, Employee_ID is a sequence of numbers and the target table contains the Employee_ID column with string data type. To generate numbers in the column, select **Override Data Type to Numeric** when you create an ad hoc generation rule.

Creating an Ad Hoc Generation Rule

You can create an ad hoc generation rule in a project. The generation techniques change based on the data type of the column that you selected. The generation parameters change based on the generation technique that you select.

1. In a project, click **Define** > **Data Generation** to access the **Data Generation** view.
2. Select a column to assign the ad hoc generation rule to.
3. Click inside the **Generation Rule** column to view the list of rules available for the data type of the column.
4. Click **New Generation Rule**.

The **New Generation Rule** window appears.



5. If you want to generate numbers in a column with string data type, select **Override Data Type To Numeric**.
6. Select the generation technique from the list.
7. Enter the generation parameters. You can choose to generate unique output values.

   Note: If a Java heap space exception occurs, contact the Administrator to increase the heap size.
8. Enter the exception test data options. Configure whether to allow null or empty spaces. Configure whether to generate data that is not valid.

Editing an Ad Hoc Generation Rule

You can edit an ad hoc rule assignment to a column.

1. In a project, click **Define** > **Data Generation** to access the Data Generation view.
2. Select a column to edit the ad hoc rule assignment.
3. Click **Actions** > **Edit Generation Rule**.

   The **Edit Generation Rule** window appears.
4. Edit the parameters that you want to change. Click **OK**.

   If there are plans that contain the generation rule assignment, the **Impacted Objects** dialog box appears with the list of affected plans.
5. To download the list of affected plans, click **Export** and save the .csv file.
6. To save the changes, click **Continue**.

   To update the changes in a plan, generate and run the plan again.

Advanced Generation Rules

Use advanced generation when you want to apply multiple generation techniques to a column or when you want to configure multiple input and output columns in the same generation rule. You can generate test data for string, numeric, and date data types.

When you configure an advanced generation rule, create input ports, output ports, and variable ports. Variable columns are work fields that you can define to temporarily store data.

When you create the ports in the generation rule, the port names do not need to be the same as the column names in the target metadata. When you assign the generation rule to columns, you must map the rule columns to the target table columns.

Create the following types of ports in an advanced rule:

Input

Optional. A port that takes input from a target column or from a parameter file. You do not need to map the input port to a column if you parameterize the input port.

Variable

A port that contains intermediate values in a calculation. The variable column receives a value from a generation technique. You can use the following generation techniques to generate data in the temporary variables: Generation Rule Set and Conditional. You can select the generation rules that are available based on the data type of the port.

You can create multiple variable ports and configure the variables based on a dependent variable port. You can configure multiple variable ports in order to combine multiple generation techniques.

Output

The target column that receives the generated data. The output column type contains an expression to generate test data.

With an advanced generation rule, you can configure generation rules with specific conditions for target columns. To generate test data in a column based on the values in another column, you can group the records based on foreign keys.

Advanced Generation Rule Example

You work for a sales department and you need to generate 100 records with 50 invoices and 50 orders. Create an advanced rule with an input column Ip_1 of string data type, a variable column Var_1, and an output column Out_1 of string data type.

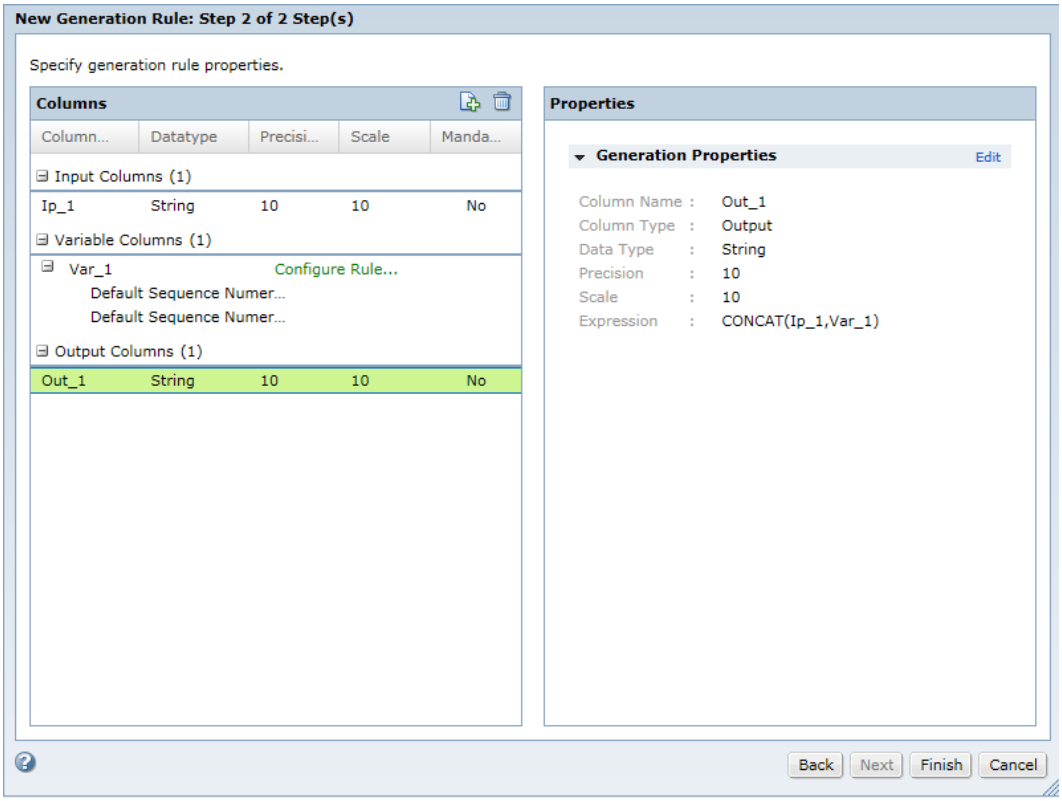
Configure the conditional rule for the variable column and select a numeric sequence generation rule for the following two conditions:

- Ip_1='Invoice-'
- Ip_1='Order-'

To concatenate the input and variable columns to create an ID column, create an output column with the following expression:`CONCAT(Ip_1,Var_1)`

Assign the advanced generation rule to a column, create a plan, and run the workflow. TDM concatenates the columns and provides 50 invoices with results Invoice-1, Invoice-2,...,Invoice-50, and 50 orders with results Order-1, Order-2, ..., and Order 50.

The following image shows the advanced rule parameters that you can configure:



Creating an Advanced Generation Rule

Create an advanced generation rule to combine more than one generation technique or to generate test data in a single column or multiple columns.

1. To access the **Policies** view, click **Policies**.
2. Click **Actions** > **New** > **Generation Rule**.

   The **Rule Wizard** appears.
3. Enter a name and an optional description for the rule.
4. Select the advanced generation rule type.
5. Click **Next**.
6. In the **Input Columns** section, click **Add New**.

   The **Add Column** dialog box appears.
7. Enter the column properties, such as the column name, data type, precision, and scale.

   You cannot enter dependencies for input columns.
8. If you want to enter a parameter name for an input column, select **Parameterize Input Port** and enter the parameter name.

   When you assign a generation rule, you need not map the input column that has a parameter value.
9. Click **OK**.
10. To enter more input columns, click **Create Port** in the **New generation Rule** dialog box.
11. In the **Variable Columns** section, click **Add New**.

12. Enter the column properties, such as the column name, data type, precision, and scale. If you have more than one variable column, you can specify the variable dependency.
13. Click **OK**.

    The variable column appears in the list of columns.
14. To configure a rule for the variable column, click **Configure Rule**.

    The New Generation Rule dialog box appears.
15. Select the generation technique. You can choose a conditional generation rule or the generation rule set.
    - Conditional. Specify conditional expressions and select the generation rules. Click **OK**.
    - Generation Rule Set. Specify the generation rules and the distribution percentages. You can also create a generation rule. To group the records based on the foreign keys, select Group record by foreign key. Click **OK**.
16. In the **Output Columns** section, click **Add New**.
17. Enter the column properties, such as the column name, data type, precision, and scale.

    You cannot enter dependencies for input columns.
18. Enter an expression for the output column. Click **OK**.
19. Click **Finish**.

Conditional Constraints

Create conditional constraints to restrict the data that you want to generate in a column based on the data in another column. The conditional constraint that you enter becomes a part of the metadata of the target table.

Configure expressions to apply constraints at the table level. Use the **Expression Builder** to enter the expressions. You can create multiple conditional constraints for a table.

You cannot use conditional constraints for the following data generation techniques:

- Numeric sequence
- Date sequence
- Random data patterns

Conditional Constraints Example

You have a table that contains airline ticket information and you want to generate airport codes in From_City and To_City columns in the table. You create and assign a dictionary generation rule to the columns. You do not want the From_City column and the To_City column to have the same value in a row. You create a conditional constraint and enter the following expression in the **Expression Builder**:

```
To_City != From_City
```

When you perform the data generation operation, TDM adheres to the conditional constraint and generates data in the From_City and To_City columns.

The following table lists sample flight details with generated data:

Flight Code	From_City	To_City
11A1	BOS	SFO
22B2	AUS	EWR
33C3	SEA	JFK

Conditional Constraints and Data Conflicts

When you perform a data generation operation, there might be conflicts between the conditional constraint and the values specified in the data generation rule parameters. The data generation operation prioritizes the data generation rule over the conditional constraint and TDM generates data with warning messages.

Conditional Constraints and Data Conflicts Example

You want to generate data in the Created_Date and Modified_Date columns in an Employee table. You create a random generation rule with a date range from January 1, 1970 to July 31, 2014, and assign the rule to both Created_Date and Modified_Date columns. You want the value in the Modified_Date column to be greater than the value in the Created_Date column. You create a conditional constraint and enter the following expression in the **Expression Builder**:

```
Modified_Date > Created_Date
```

When the Created_Date column contains July 31, 2014, TDM cannot generate data that adheres to the conditional constraint because the data generation rule parameters restrict the end dates to July 31, 2014. TDM generates a date within the specified range in the Modified_Date column and generates warning messages.

The following table lists sample employee details with generated data:

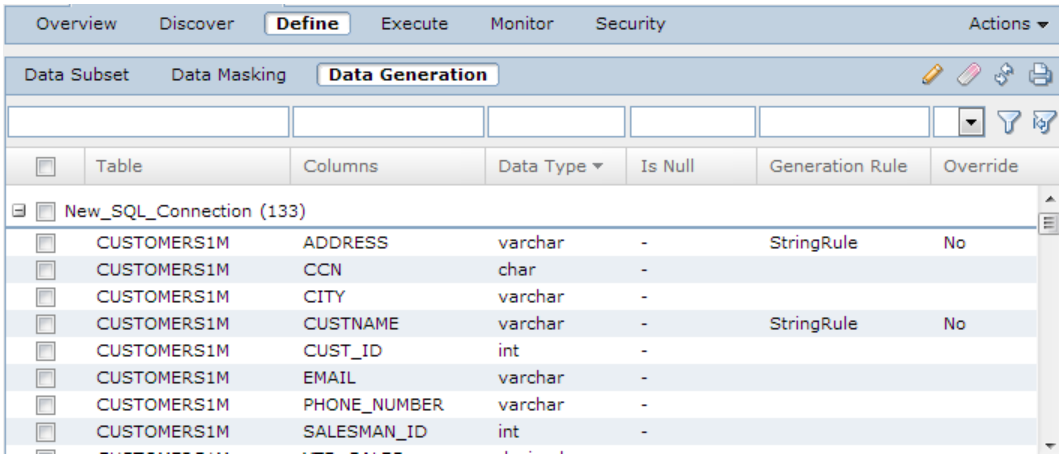
Employee ID	Created_Date	Modified_Date
1111	7/31/2014	1/13/2002
2111	6/5/2000	5/6/2010
8112	7/31/2014	5/16/1990

Data Generation Rule Assignments

You can assign a global or an ad hoc generation rule to a column within a project.

To assign a rule, choose a rule from the list in the **Generation Rule** column or select **New Generation Rule**. If you do not select a rule, the column receives the default rule that you configure in default settings.

The following image shows the **Define** | **Data Generation** view in a project:



The following table describes the fields in the **Data Generation** view:

Column	Description
Table	Name of the table.
Column	Name of the column of the generated data.
Data Type	Data type of the column of the generated data.
Is Null	Indicates whether the column can contain null values.
Generation Rule	The generation rule that you want to apply to the column. When you click inside the column, you can choose which rule to apply to the column. The rules change based on the data type of the column. Select **New Generation Rule** to create an ad hoc generation rule for the column.
Override	Shows the override property status for a rule. If the property is **Yes**, you can override the rule properties when you assign the rule to a column. When you override the rule parameters for a column the **Override** column value is **Yes-Overridden**.

Available Data Generation Rules

The **Data Generation** view lists the rules that you can assign to each column of the target table.

If no global rules appear for a column when you click **Generation Rule**, make sure that the project contains global generation rules for the column datatype.

Ad hoc generation rules are available for columns of all datatypes. Click **New Generation Rule** to create an ad hoc generation rule.

Note: In the **Data Generation** view, you must click the **Save** icon each time you assign a rule. If you do not save the rule assignment before you assign the next rule, the Test Data Manager discards the rule assignment.

Assigning Standard Generation Rules to Columns

Assign standard generation rules to one or more target columns to generate data in the columns.

1. In a project, click **Define** > **Data Generation** to access the **Data Generation** view.

2. Select a column to assign a generation rule to.
3. Click inside the **Generation Rule** column to view the list of rules available for the data type of the column.
4. Select a rule.
5. Click **Save** for each column that you update.

   If there are plans that contain the generation rule assignment, the **Impacted Objects** dialog box appears with the list of affected columns, plans, and data domains.
6. To download the list of affected objects, click **Export**, and save the .csv file.
7. To save the updates, click **Continue**.

   To update the changes in a plan, you must generate and run the plan again.

Assigning Custom Generation Rules to Columns

Assign custom generation rules to one or more target columns to generate data in the columns.

1. In the **Define | Data Generation** view, click the **Generation Rule** column for the column that requires a mapplet rule.
2. Select the mapplet rule from the list.

   The **Custom Rule Assignment** dialog box appears with the list of columns in the table.
3. Select a source input column in the left panel and a rule input port in the right panel.

   Note: When you select a source input column, the rule ports with same datatype become available for mapping.
4. Click the **Link** icon to create a mapping.

   An arrow appears that links the source port to the rule port.

   Note: To create a mapping, you can also click **Show Simple** and map the ports.
5. If you need to remove the link, select the source input port and the rule input port. Click the **Unlink** icon.

   The arrow between the ports disappears.
6. After you map the rule input ports, click **Next**.
7. Select a rule output port in the left panel and a source input column in the right panel.
8. Click the **Link** icon to create a mapping. To remove a mapping, click the **Unlink** icon.
9. Click **Save**.

Assigning Advanced Generation Rules to Columns

Assign advanced generation rules to one or more target columns to generate data in the columns.

1. In the **Define | Data Generation** view, click the **Generation Rule** column for the column in which you want to assign the advanced rule.
2. Select the advanced generation rule from the list.

   The **Advanced Rule Assignment** dialog box appears with the list of table columns in the left panel and rule columns in the right panel.
3. Select a table input column from the left panel and a rule input column from the right panel.

   Note: When you select a table input column, the rule ports with same data type become available for mapping.
4. Click the **Link** icon to create a mapping.

An arrow appears that links the source port to the rule port.

Note: To create a mapping, you can also click **Show Simple** and map the ports.

5. If you specified a parameter input port or if you do not have input ports in the advanced rule, you do not need to map the input columns.
6. If you need to remove the link, select the source input port and the rule input port. Click the **Unlink** icon.

   The arrow between the ports disappears.
7. After you map the rule input ports, click **Next**.
8. Select a rule output port in the left panel and a table output column in the right panel.
9. Click the **Link** icon to create a mapping. To remove a mapping, click the **Unlink** icon.
10. If you configured to group the child records based on a foreign key in the variable column of the advanced rule, you must select the foreign key column.
11. Click **Save**.

Assigning Auto Rules to Columns

You can assign an auto rule to target columns if the Inferred Rule status is Yes. You can edit the inferred rule properties after you assign auto rule to the target.

1. In a project, click **Define** > **Data Generation**.
2. Select a table column on which you want to assign the rule.
3. Click inside the **Generation Rule** column and select **Auto Rule**.
4. To save the assignment, click **Save**.
5. To view the assigned inferred rule properties, select the column. To update the rule properties, click **Edit**.
6. To assign Auto Rule to multiple columns, select multiple rows and click **Generation Rule Assignment**.

   The **Apply Generation Rules** dialog box appears.
7. Select **Auto Rule**, and click **Apply** to save the assignment.

   Note: You must not apply an auto rule to a primary key column because a primary key column cannot contain null values.

Deleting Generation Rule Assignments

You can delete a generation rule assignment to a column.

1. In a project, click **Define** > **Data Generation**.
2. Select the column that you want to delete the rule assignment for.
3. Click **Actions** > **Clear Assignment**.

   If the rule assignment does not exist in a plan, TDM deletes the generation rule assignment. If there are plans that contain the generation rule assignment, the **Impacted Objects** dialog box appears with the list of affected columns, plans, and data domains.
4. To delete the generation rule assignment, click **Continue**. To download the list of affected objects, click **Export**, and save the .csv file.

   To update the changes in a plan, you must generate and run the plan again.

Data Generation for XSD Sources

You can apply data generation rules to XML elements and attributes to generate test data.

View and open the XSD file from the **Discover | Files** view of the project. You can view the details of the XML elements and attributes in the XSD map view or the grid view.

Select an element, and apply a data generation rule to the element. You can select a link between the elements and specify the minimum and maximum number of child records for each parent to generate test data.

You can select multiple elements and assign a default generation rule to the selected elements. You can remove all the default generation rule assignments, and assign another generation rule.

You can assign standard and custom generation rules to generate test data in XSD sources. You cannot use reference lookup and effective dates generation rules for an XSD source.

Data Generation Task Flow for XSD Sources

To generate data from XSD sources, you can assign the generation rules to the XML elements and attributes.

Perform the following high-level tasks to generate test data:

1. Create a project.
2. Import the XML source definition from the PowerCenter repository.
3. Add data generation rules to the project.
4. Open the XSD file from the project.
5. Select an element or an attribute. Specify a generation rule for a data domain or apply generation rules individually.
6. Create a data generation plan and add the XSD file. You can add multiple XSD files to a plan. Define the data generation parameters, and specify the target directory.
7. Generate a workflow from the plan.
8. Run the workflow, and monitor the progress of the workflow.
9. Verify that the data is generated in the target.

Applying Data Generation Rules to XML Elements and Attributes

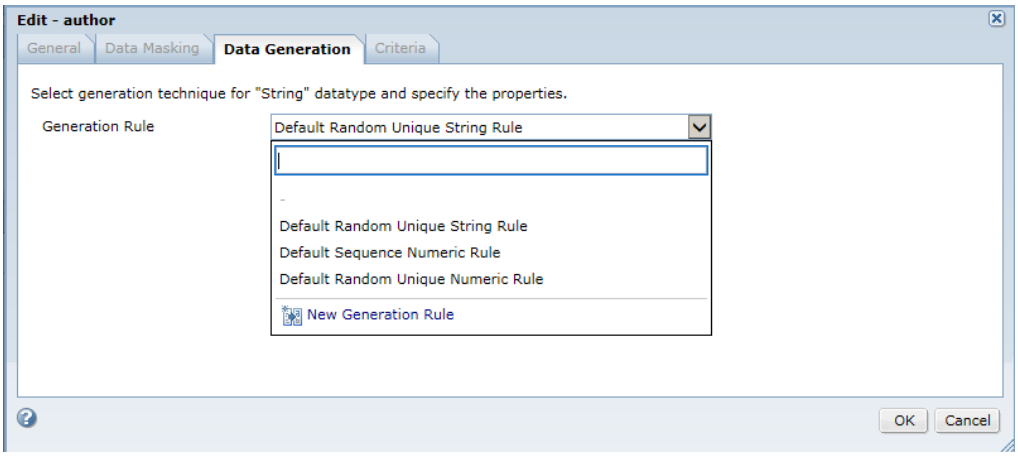
Apply data generation rules to XML elements and attributes to generate test data in an XML file.

1. In a project, click **Discover** > **Files**.

   A list of XSD files in the project appears.
2. Click the XSD file.

   The XSD file opens in a separate tab.
3. Select the elements and attributes on which you want to apply the generation rule.
4. In the **Details** pane, click **Edit** in the **Data Generation** section.
5. Assign a data generation rule to the selected element or attribute. You can also create and assign an ad hoc generation rule.

The following image shows a sample data generation rule assignment to an XML element:



6. To save the assignment, click **OK**.
7. To assign default rules to multiple objects, click **Default Assignment**.
8. To clear rule assignments, click **Clear Assignment**.
9. If you select a link between the attributes or elements, click **Edit** from the **General Properties** section. Enter the minimum and maximum number of child records, and click **OK**.

   Note: If you specify the range of child records for each link, TDM prioritizes the parent child ratio configured at the individual relationship level over the plan-level configuration.

Data Generation Plans and Workflows

A data generation plan contains XSD files, tables, and entities that represent the data that you want to generate.

The plan includes the components that have generation rule assignments. You generate a workflow from a plan to perform the generation operations. When you start a workflow for the plan, the data generation operation starts.

CHAPTER 10

Data Generation Techniques and Parameters

This chapter includes the following topics:

- Data Generation Techniques and Parameters Overview, 199
- Data Generation Techniques, 200
- Data Generation Parameters, 201
- Custom Generation, 202
- Advanced Generation, 203
- Dictionary Generation, 206
- Effective Dates Generation, 208
- Expression Generation, 210
- Random Generation, 211
- Credit Card Number Generation, 217
- Reference Lookup Generation, 220
- Sequence Generation, 221
- Set of Values Generation, 223
- Conditional Generation, 225

Data Generation Techniques and Parameters Overview

A data generation technique defines how you want to generate data in a column. A generation parameter is an option that you configure for the technique.

The type of generation technique that you can apply depends on the datatype of the column that you need to generate. When you choose a generation technique, Test Data Manager displays parameters for the generation technique.

Some generation techniques are available only for ad hoc generation rules. The generation techniques that you can select to create a rule can change when you create an ad hoc generation rule.

You can create generation rules that create random strings, use data from a dictionary or set of values, or that create data based on a sequence that you define.

Data Generation Techniques

You can apply generation techniques based on the target datatype that you configure for a column.

For example, if the column datatype is numeric, you can define generation values that are within a fixed range or values in a sequence. You can select an additional rule to create values that are not valid values within the data set.

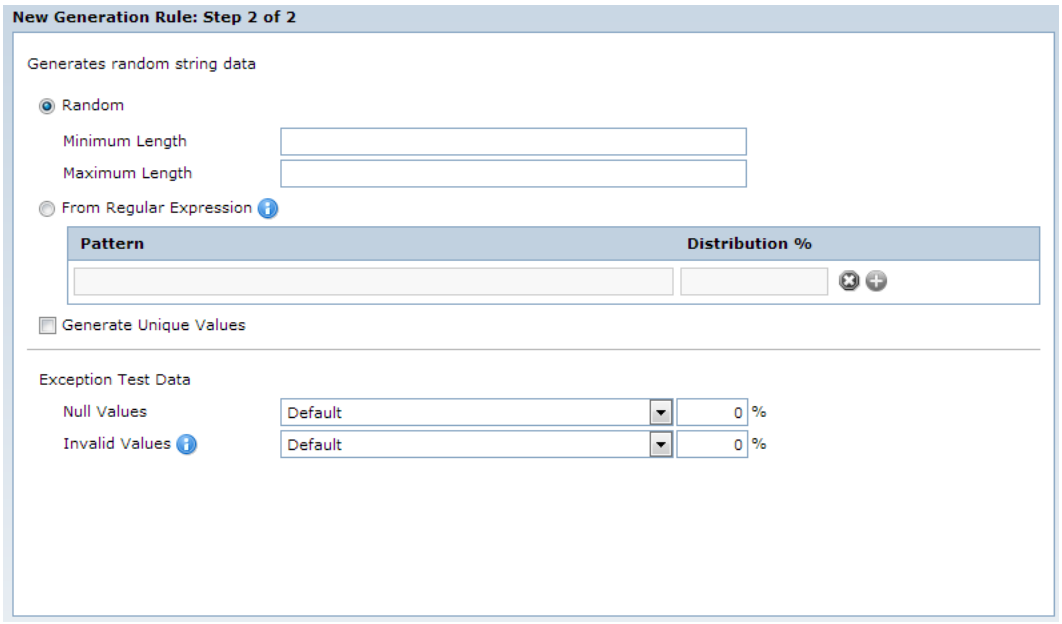
The following table describes the generation techniques that you can choose when you define a rule:

Generation Technique	Description
Custom	A technique that applies generation rules from a mapplet. The custom mapplet accepts input and uses the logic in the mapplet to generate the output.
Advanced	A combination of generation techniques that generates test data in target columns based on the configuration of one or more than one input port, variable port, and output port.
Dictionary	Standard technique that imports dictionary values to generate data.
Effective Dates	Ad hoc technique that generates related dates for two columns in the target database.
Expression	Ad hoc technique that accepts expression as the input to generate the output.
Random	Standard technique that generates random strings, values, and dates.
Credit Card	Standard technique that generates credit card numbers based on the type of the card.
Reference Lookup	Ad hoc technique that generates data from a reference lookup table. You can have multiple column assignments within a reference lookup rule.
Sequence	Standard technique that generates numeric and date values in a sequence.
Set of Values	Standard technique that uses a finite set of values to use in data generation.
Conditional	Ad hoc rule in which you can specify a conditional expression and a generation rule to generate test data.

Data Generation Parameters

Configure data generation parameters to define how to apply a data generation technique. The parameters that you configure depend on the data type that you need to generate. Some generation techniques are not available for all data types.

The following image shows data generation parameters that appear when you configure a random string generation rule:



Exception Test Data

Configure exception test data parameters to generate null and values that are not valid.

Select a global rule that you previously saved or select default to generate random null values and values that are not valid. Invalid values are data values that do not conform to the generation rule.

For example, you want to create a random date generation rule that generates 85% of the dates between 2012, and 15% of the dates between 2005 and 2010. You create a rule called Dates_2005 that generates dates from 2005 to 2010. You save the Dates_2005 rule. You create a rule called Recent_Dates that generates dates from 2012 to 2013. When you create the Recent_Dates rule, you select Dates_2005 in the invalid values parameter and enter 15 in the percent parameter.

You cannot generate null values for MongoDB sources.

Exception Test Data Parameters

You can configure exception test data parameters based on the data generation rules that you create. You can use exception test data parameters to generate null values and values that are not valid.

The following table describes the exception test data parameters:

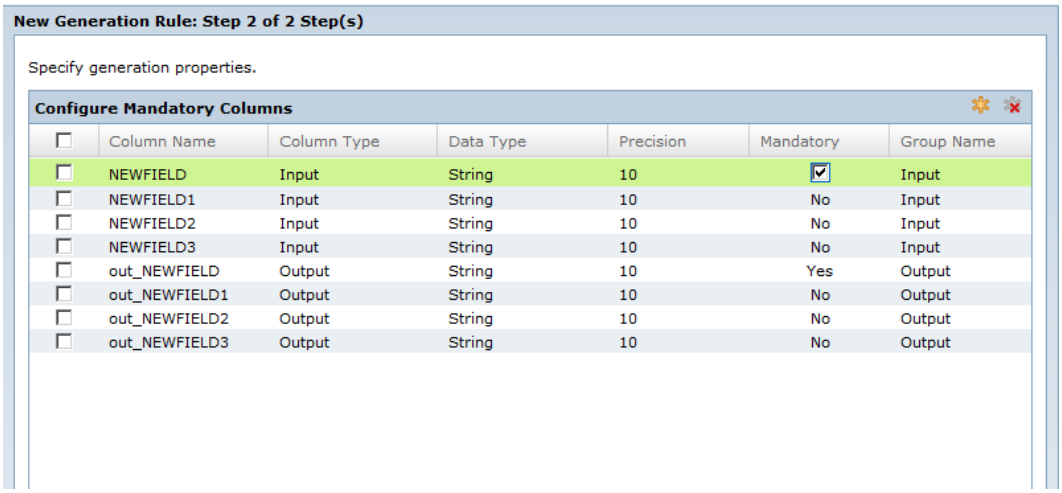
Parameter	Description
Null Values	A list of global rules that you previously created or the default rule to generate null values. The rules are available based on the data type you select. Percentage defines the percentage of rows that can contain the null values. You can enter a whole number or a decimal number up to two places.
Invalid Values	A list of global rules that you previously created or the default rule to generate values that are not valid. The rules are available based on the data type you select. Percentage defines the percentage of rows that can contain the values that are not valid. You can enter a whole number or a decimal number up to two places.

Custom Generation

Create custom generation rules to generate data for string, numeric, and date data types.

Custom generation rule is a global rule. To create custom generation rule, import a PowerCenter mapplet. Mark at least one input and one output column as required.

The following image shows the custom generation parameters:



Column Name	Column Type	Data Type	Precision	Mandatory	Group Name
NEWFIELD	Input	String	10	☑	Input
NEWFIELD1	Input	String	10	No	Input
NEWFIELD2	Input	String	10	No	Input
NEWFIELD3	Input	String	10	No	Input
out_NEWFIELD	Output	String	10	Yes	Output
out_NEWFIELD1	Output	String	10	No	Output
out_NEWFIELD2	Output	String	10	No	Output
out_NEWFIELD3	Output	String	10	No	Output

Custom Generation Example

An EMPLOYEES table contains columns for first name and last name. Create and assign standard generation rules to generate first names and last names. Create a custom generation rule to combine the first and last name of an employee to generate the full name in the target table.

Custom Generation Parameters

Configure input and output columns that you import from a mapplet when you create a custom generation rule.

The following table describes the general properties that you can configure for input and output columns:

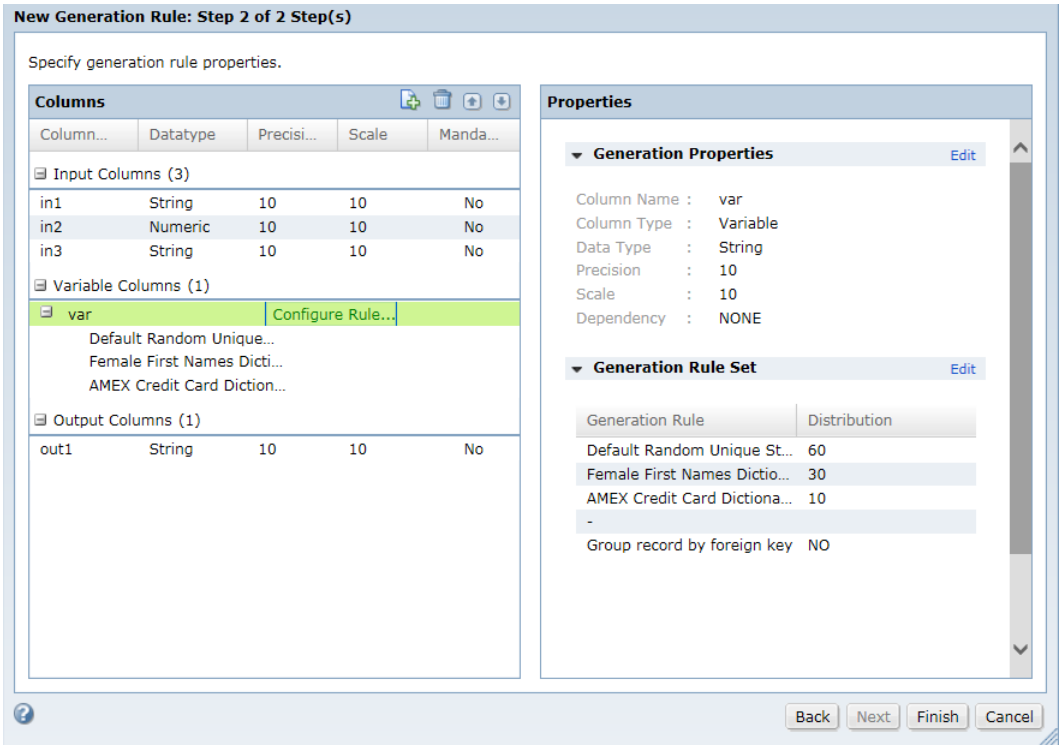
Parameter	Description
Column Name	The name of an input or an output column within a mapplet. When you assign the rule to a column, you map the column names in the rule to column names in the database.
Column Type	The column type. The mapplet can contain the following types of columns: - Input. Receives the input data. - Output. Returns the output data.
Data Type	The data type of the column.
Precision	The precision for the column. The maximum number of digits or the maximum number of characters that the column can accommodate. For example, 874.560 has a precision of 6.
Mandatory	Indicates if you must assign the column to a table column in the project. Applies to input and output columns. You must set at least one input and one output column as required.
Group Name	Specifies the group to which the columns belong. The group name can be input, output, or any other name that you provide when you create a mapplet.

Advanced Generation

An advanced generation rule is a combination of generation techniques that generates test data in target columns based on the configuration of one or more than one input port, variable port, and output port.

You can generate test data for string, numeric, and date data types. In an advanced rule, create input columns, variable columns, and output columns based on the test data requirements. Configure a generation rule for a variable column, and add an expression for the output column.

The following image shows the advanced generation rule parameters that you configure:



Advanced Generation Example

You want to generate email IDs for the employees in EMAIL_ID column in the EMPLOYEE table. Create an input port in_FirstName of string data type. Create a variable port var_Domain of string data type. Configure the Generation Rule Set technique for the variable port. Configure two rules to generate yahoo and gmail domain names. Create an output port out_Email to concatenate the input name and the variable domain with the following expression:

```
CONCAT(CONCAT(in_FirstName,'@'),var_Domain)
```

Assign the rule to the EMAIL_ID column, and generate and run the workflow to produce the output email IDs.

The following table shows a sample output:

First_Name	EMAIL_ID
Richmond	Richmond@gmail.com
Debora	Debora@gmail.com
Shaniah	Shaniah@yahoo.com
Lena	Lena@yahoo.com

Advanced Generation Parameters

Configure parameters for each column that you create in an advanced generation rule.

The following table describes the general properties that you can configure for input, output, and variable columns:

Parameter	Description
Column Name	The name of an input, output, or variable column. Enter a name. The name does not have to match the name of a column in the source. When you assign the rule to the target metadata in a project, you map the column names in the rule to column names in the database.
Column Type	The column type. You can configure the following types of columns: - Input. Receives the target metadata. - Variable. A temporary column that contains intermediate values. You can apply generation rules to variable column values in order to mask data before returning data to output columns. - Output. Returns the output data. You can apply an expression to a variable column and return the data in the output column.
Datatype	The data type of the column. Select string, numeric, or date data type.
Precision	The precision for the column. The maximum number of digits or the maximum number of characters that the column can accommodate. For example, 798.650 has a precision of 6.
Scale	Number of digits to the right of the decimal point in a number.

The following table describes the properties that you can configure for input columns:

Parameter	Description
Parameterize Input Port	Optional. Use a parameter value for the input column. You do not need to map the input port to a column if you parameterize the input port.
Parameter Name	The name of the parameter that you can use for the input column. Required if you choose to parameterize the input port.

The following table describes the masking properties that you can configure for variable columns:

Parameter	Description
Dependency	Optional. Select a variable port as a dependent port for another variable port. When you create a second variable port, you can see the first variable port in the drop-down list.
Generation Technique	Applies a generation rule to the input column and writes the results in the variable column. You can create multiple variable columns and configure rules. You can select the following techniques: - Conditional. Generates string data with a condition. Specify a conditional expression based on which you can generate test data in a target column. Select a generation rule that you can apply to the target column based on the specified condition. - Generation Rule Set. The data generation rule to apply to the input column. Specify a distribution percentage. You can group the records based on a foreign key.

The following table describes the properties that you can configure for output columns:

Parameter	Description
Expression	Required. An expression to apply to input and variable columns to generate an output. You can create the expression in the Expression Builder.

Dictionary Generation

A dictionary generation rule is a global or ad hoc generation rule that loads dictionary values into the target test database.

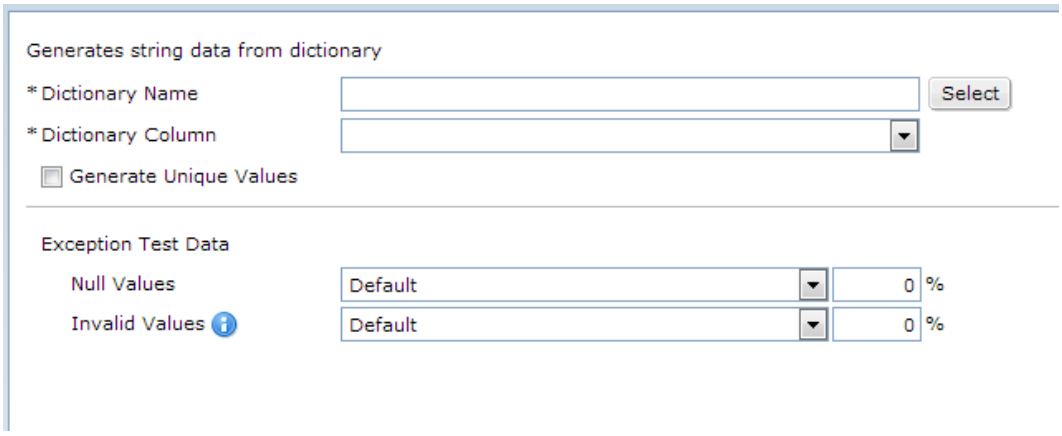
Before you create a dictionary generation rule, you must add a dictionary to TDM in the **Administrator | Dictionaries** view. Users that have permissions to create generation rules have permissions to add dictionaries.

Dictionary generation can use the string, numeric, date, and binary data types. You can create a rule based on a dictionary column. The binary data type contains a serial number column that increases performance. Generate dictionary values for columns that have a finite number of values, such as states or countries.

You can create dictionary generation rule either from a flat file or relational database. You can generate data for a binary data type from relational databases but not from flat files.

Dictionary generation rules can be global rules or ad hoc rules.

The following image shows the dictionary generation parameters:



Dictionary Generation Example

You want to load customer names into the target database. You import a dictionary that has first and last name columns. You create two dictionary rules with the string data type. When you create the first name rule, you select the dictionary that contains the names and choose the first name dictionary column. In the last name rule, you select the same dictionary and choose the last name dictionary column.

Dictionary Generation Parameters

Create a dictionary generation rule to load dictionary values into target databases. Configure dictionary generation parameters for string, date, numeric, and binary data types.

The following table describes the parameters that you can configure to generate dictionary data:

Parameter	Description
Dictionary Name	The name of the dictionary. You need to import a dictionary in TDM.
Dictionary Column	The column that you want to get the data from. If the values in the dictionary are larger than the size of the column then it truncates the values.
Dictionary Lookup Column	The lookup column in the dictionary that you can match with the table column to generate data. When you assign a standard generation rule to a column or create an ad hoc generation rule, you can specify the lookup column. This field is optional. If you enter double data type value in the lookup column and table column, the process generates null values in the target table.
Table Column	The column in the target table based on which you can match the lookup column in the dictionary. When you assign a standard generation rule to a column or create an ad hoc generation rule, you can specify the table column. This field is optional. If you enter double data type value in the lookup column and table column, the process generates null values in the target table.
Serial Number Columns	A serial number to find the location of the data. You can configure serial number for relational databases and not for flat files. You enter the serial number column for binary data types.
Generate Unique Values	Generates unique data values. You cannot generate unique values for flat file dictionaries and binary data types.
Null Values	A list of global rules or a default rule that is available for the data type you select. A null value is a value that a default rule or a global rule generates in the target for the columns based on the data type you select.
Invalid Values	A list of global rules or a default rule that is available for the data type you select. Any value that does not exist in the dictionary is not valid. You cannot generate values that are not valid for binary data types.

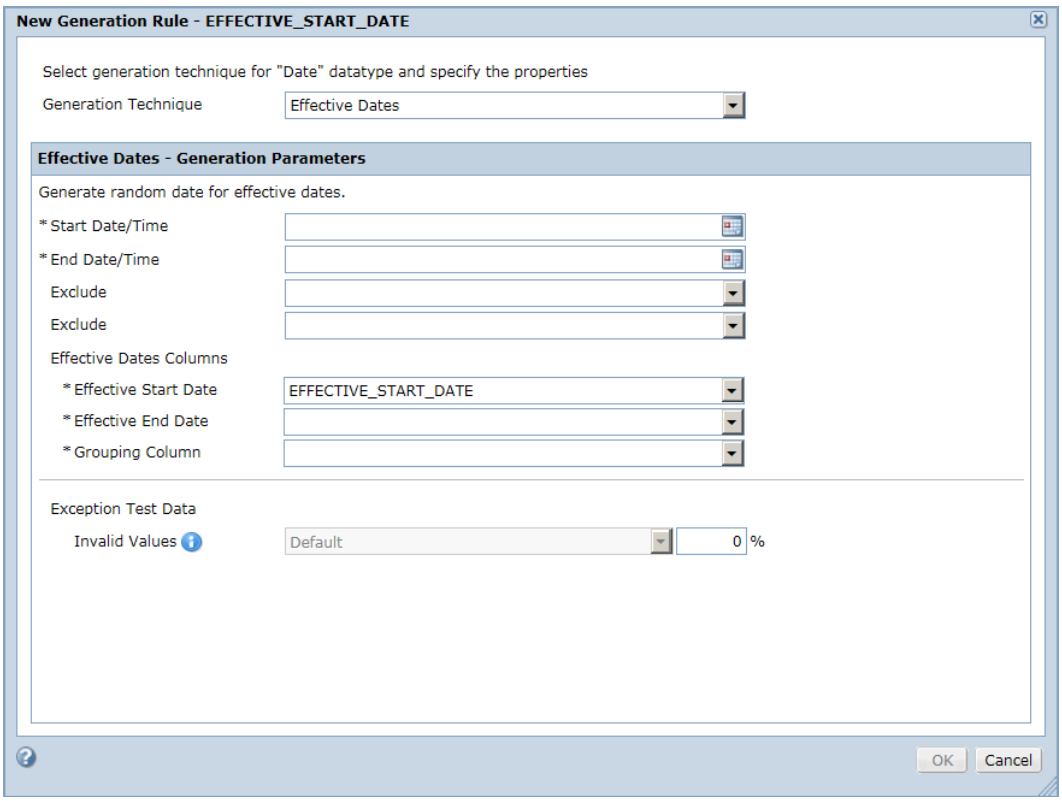
Effective Dates Generation

An effective dates generation rule is an ad hoc rule that generates related date data for two columns. The effective dates are sequences of dates that consist of start dates and end dates. You can create effective dates rules with the date data type.

An effective dates generation rule generates a random start date within a range. The rule creates an end date that is after the start date and another start date that immediately follows the end date. The number of rows in the grouping column that have the same value determines the number of dates in the effective dates sequence.

The column in which you create the rule must be either the effective start date or the effective end date. If you create the rule on the effective start date column, you must select a different column as the effective end date column. If you create the rule on the effective end date column, you must select a different column as the effective start date column. The rule generates date values for both columns. You select the grouping column to generate sequences of dates in multiple rows.

The following image shows the effective dates generation parameters:



Effective Dates Generation Example

You want to generate start dates and end dates for customer addresses.

The target table has two customers. Each customer has one identification number. The table has three address rows for each customer. You choose a date range to generate a random effective start date for each customer. You select an effective start date column to put the start date into. You select an effective end date column to put an end date into. You select the customer identification number as the grouping column.

The rule generates a random start date for the first address row for each customer. The rule generates a random end date that occurs after the start date. The rule generates a start date for the next row that is after the end date of the previous row.

The following table shows a sample table with the generated data:

Customer ID	Address	Start Date	End Date
1111	55 Crestwood Dr	11/20/2001	06/17/2003
1111	200 Larkin St	06/18/2003	04/16/2004
1111	1000 Cooke Ln	04/17/2004	08/16/2012
2222	12708 Danbrooke Ct	03/06/2008	05/19/2010
2222	1200 S Sixth Ave	05/20/2010	07/06/2011
2222	2243 Creeksie Ct	07/07/2011	11/24/2013

Effective Dates Generation Parameters

Create effective dates generation rule for the date data type.

The following table describes the parameters that you can configure to generate data for effective dates:

Parameter	Description
Start Date/Time	The date and time at which you can start to generate dates.
End Date/Time	The date and time at which you can stop to generate dates.
Exclude	The days of the week that you do not want to include in the generated dates. You can enter up to two days, and both the days cannot be same.
Effective Start Date	The column name in the target table where you want to put the generated effective start date. Default is the column name in which you create the effective dates generation rule. You can change it to any other available column name.
Effective End Date	The column name in the target table where you want to put the generated effective end date. You can select any column other than the effective start date.

Parameter	Description
Grouping Column	The column that determines how you group the data to generate effective dates in the target columns.
Invalid Values	Any value with effective end date greater than the effective start date. If effective start date is less than effective end date of the previous record for the grouping column, the values are not valid. This parameter has default rule selected and you cannot change the rule. You enter the percent for values that are not valid.

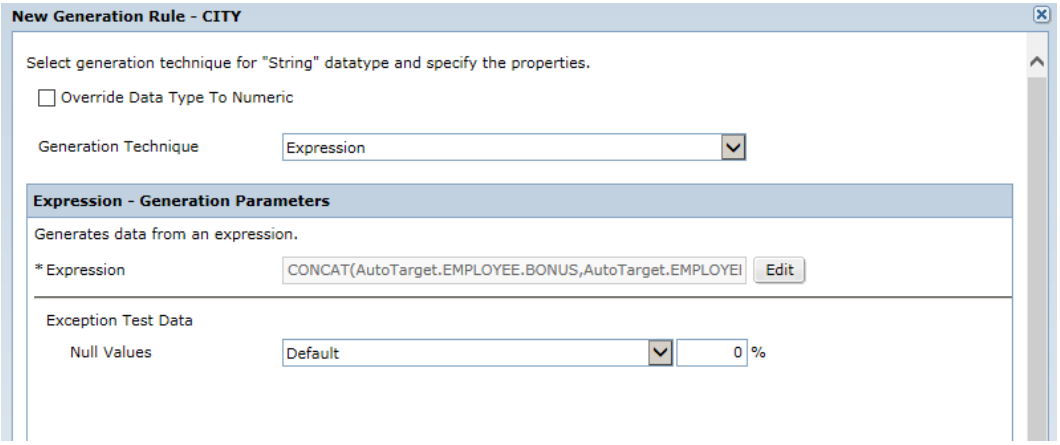
Expression Generation

An expression generation rule is an ad hoc rule that generates data from an expression. You can create expression rules with the string, numeric, and date datatypes.

When you create an expression generation rule, you use the **Expression Builder** to create an expression. In the **Expression Builder**, you can choose from a list of columns and functions to include in the expression. To enter a join condition, you can choose columns and operators. Click the **Add** arrow to include the column in the expression.

If you select **Override Data Type to Numeric** for a column with string data type, you cannot enter a numeric expression.

The following image shows the expression generation parameters:



Expression Generation Example

You want to create a Name column that combines the data from the FirstName and LastName columns. You use the following expression to create the rule:

```
CONCAT (FirstName+'/'+LastName)
```

Expression Generation Parameters

You can configure expressions to generate data for string, date, and numeric data types.

The following table describes the parameters that you can configure to generate expressions:

Parameter	Description
Expression	An expression that you can configure to generate test data. If you select to override a string data type column with numeric data type, you cannot enter expression to generation numbers.
Null Values	A list of global rules or default rule that are available for the data type you select. A null value is value that a default rule or a global rule generates in the target column based on the datatype you select.
Invalid Values	A list of global rules or default rule that are available for the data type you select. Any value that does not follow the specified expression is not valid.

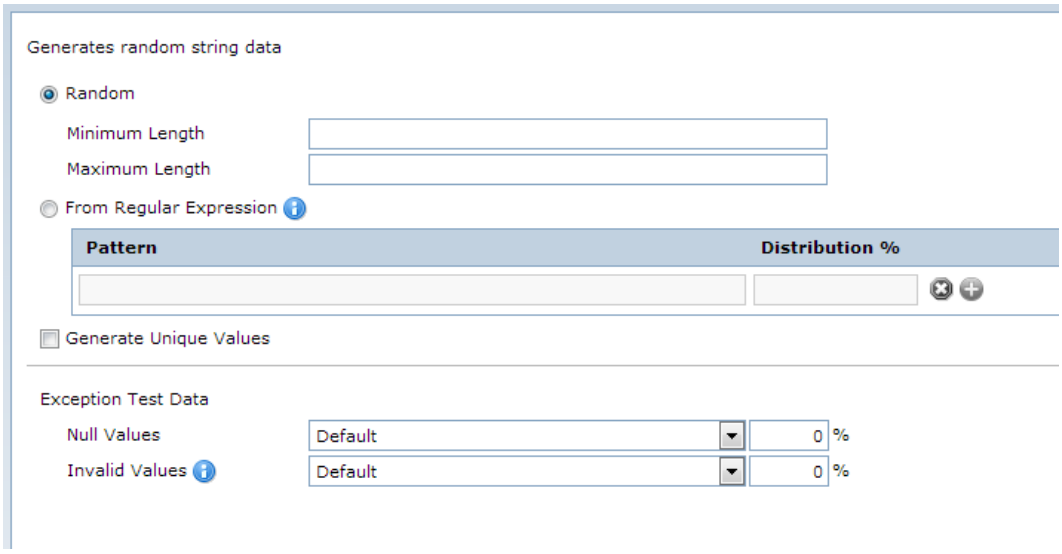
Random Generation

A random generation rule is a global or ad hoc rule that creates random string, numeric, and date values. You can use random generation to generate data such as salary and birth dates.

When you configure random generation, you can specify a minimum and maximum length for data strings and a minimum and maximum value for numeric data. You can include a regular expression to create a fixed percentage of the generated data. The percentages of the pattern distributions must total 100. When you generate dates, you can select a start date and end date to create random dates within a range.

Random generation rules can be global rules or ad hoc rules.

The following image shows the random generation parameters:



Random Generation Example

You want to generate addresses for customers. You create a rule that uses the random technique and the string data type. You define a regular expression in the rule. The regular expression creates rows with a four digit number, a random string of 12 characters, and "St." or "Ave." You set the pattern percentage to 100 to generate all the rows with the regular expression.

Date Random Generation Parameters

Create a random data generation rule for date data types.

The following table describes the parameters that you can configure to generate random data for date data types:

Parameter	Description
Start Date/Time	The date and time at which you can start to generate random dates.
End Date/Time	The date and time at which you can stop to generate random dates.
Exclude	The days of the week that you do not want to include in the generated dates. You can enter up to two days, and both the days cannot be same.
Null Values	A list of global rules or default rule that are available for the date data type. A null value is a value that a default rule or a global rule generates in the target column for the date data type.
Invalid Values	A list of global rules or default rule that you can select for the date data type. Any value that is less than the minimum or more than the maximum for a specified date range is not valid. Any date that falls on excluded dates is not valid.

String Random Generation Parameters

Create a random data generation rule for string datatypes.

The following table describes the parameters that you can configure to generate random data for string datatype:

Parameter	Description
Random	A range provided to generate random data. You can select Random to specify the range of the string for data generation.
Minimum Length	The minimum length for the string datatype. Default is 1.
Maximum Length	The maximum length for the string datatype. Default is the maximum value that the column permits.
From Regular Expression	A regular expression to generation random data. You can select From Regular Expression to create fixed percentage of data for generation.
Pattern	A regular expression that describes the format of generated data.
Distribution	The percentage of rows with the regular expression data. If you want to use more expressions, you can add more rows.
Generate Unique Values	Generates unique data values. You cannot generate unique values when you select From Regular Expression.
Null Values	A list of global rules or default rule that are available for the string datatype. A null value is a value that a default rule or a global rule generates in the target columns for the string datatype.
Invalid Values	A list of global rules or default rule that you can select for the string datatype. Any value that is less than the minimum length or more than maximum length for a specified range is not valid. A string that does not match the specified pattern is not valid. You cannot generate values that are not valid when you select From Regular Expression.

Numeric Random Generation Parameters

Create random data generation rule for numeric datatypes.

The following table describes the parameters that you can configure to generate random data for numeric datatype:

Parameter	Description
Random	A rule to generate random data. You can select Random to specify the range of numeric values for data generation.
Minimum Value	The minimum value for the numeric datatype. Default is 0.
Maximum Value	The maximum value for the numeric datatype. Default is the maximum value that the column permits.
From Regular Expression	A regular expression to generation random data. You can select From Regular Expression to create fixed percentage of data for generation.
Pattern	A regular expression that describes the format of generated data.
Distribution	The percentage of rows with the regular expression data. If you want to use more expressions, you can add more rows.
Generate Unique Values	Generates unique data values. You cannot generate unique values when you select From Regular Expression.
Null Values	A list of global rules or default rule that are available for the numeric datatype. A null value is a value that a default rule or a global rule generates in the target for the numeric datatype.
Invalid Values	A list of global rules or default rule that you can select for the numeric datatype. Any value that is less than the minimum or more than the maximum for a specified range is not valid. A value that does not match the specified pattern is not valid. You cannot generate values that are not valid when you select From Regular Expression.

Data Patterns for Random Generation

You can enter data patterns from regular expressions to generate string and numeric data.

To generate numbers that contain special characters or any other operators, you use random string data generation technique. You can use the following operators to generate string data patterns: `. , \d , \w, (opt1| opt2|…..), {} , []`.

To generate numbers that do not contain special characters or any other operators, you use random numeric data generation technique. To generate numeric data, you can combine the following patterns: `\d, alternates (1|2|3|…), and [0-9]`. You cannot nest the alternates.

When you enter data patterns to generate the credit card number, Social Security number, and Social Insurance numbers, the generated data might not be valid. These numbers follow certain algorithms and you cannot use data patterns to generate valid numbers.

String Data Pattern Examples

You want to generate email addresses, credit card numbers, phone numbers, Social Insurance numbers, Social Security numbers, dates, IP addresses, and employee IDs from regular expressions.

Email Address

To generate email addresses, you must enter the pattern in the following order:

1. Enter the first eight characters as lowercase alphabet characters [a-z].
2. Enter the sign (@).
3. Enter hotmail, yahoo, gmail, aol, or Comcast.
4. Enter .com, co.uk, or .net.

To generate an email address, use the following pattern: `[a-z]{8}@(hotmail|yahoo|gmail|aol|Comcast)\.(com|co\.uk|net)`. To generate a data pattern for uppercase and lower alphabet characters, you can enter `([a-z]|[A-Z]){8}@(gmail|hotmail)\\.(com|co\\.uk)`.

Credit Card Number

To generate a credit card number, you enter the first character and the pattern.

The following table shows the output for sample patterns that generate credit card numbers:

Pattern	Sample Output
4\d{3}-\d{4}-\d{4}-\d{4}	4312-0036-5436-0876
3\d{4} \d{4} \d{6}	3852 0000 023237

Phone Number

The following table shows the output for sample patterns that generate phone numbers:

Pattern	Sample Output
\d{3}\.\d{3}\.\d{4}	617.555.1212
\d{3}-\d{3}-\d{4}	617-555-1212
\(\d{3}\) \d{3}-\d{4}	(617) 555-1212
\+\d\.\d{3}\.\d{3}\.\d{4}	+1.617.555.1212
\d\.\d{3}\.\d{3}\.\d{4}	1.617.555.1212

Social Insurance Number

The following table shows the output for sample patterns that generate Social Insurance numbers:

Pattern	Sample Output
\d{3}-\d{3}-\d{3}	289-645-236
\d{3}\.\d{3}\.\d{3}	289.645.236
\d{3} \d{3} \d{3}	289 645 236

Social Security Number

The following table shows the output for a sample pattern that generates a sample Social Security number:

Pattern	Sample Output
\d{3}-\d{2}-\d{4}	999-99-9999

Date

The following table shows the output for sample patterns that generate dates:

Pattern	Sample Output
(1|3|5|7|8|10|12)-([1-9]|[12][0-9]|30|31)-(200[0-9]| 201[0-3])	5-30-2013
(4|6|9|11)-([1-9]|[12][0-9]|30)-(200[0-9]|201[0-3])	11-22-2012
(0[1-9]|1[0-2])\/((0[1-9]|2\d)|3[0-1])\/(19\d\d|200[0-3])	12/31/2003

IP Address

The following table shows the output for sample patterns that generate IP addresses:

Pattern	Sample Output
([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])\.([0-9]|[1-9][0-9]|1[0-9]{2}|2[0-4][0-9]|25[0-5])	255.255.0.0
(25[0-5]|2[0-4][0-9]|[0-1]{1}[0-9]{2}|[1-9]{1}[0-9]{1}|[1-9])\.(25[0-5]|2[0-4][0-9]|[0-1]{1}[0-9]{2}|[1-9]{1}[0-9]{1}|[1-9]|0)\.(25[0-5]|2[0-4][0-9]|[0-1]{1}[0-9]{2}|[1-9]{1}[0-9]{1}|[1-9]|0)\.(25[0-5]|2[0-4][0-9]|[0-1]{1}[0-9]{2}|[1-9]{1}[0-9]{1}|[0-9])	192.168.0.1

Employee ID

The following table shows the output for sample patterns that generate employee IDs:

Pattern	Sample Output
([A-Z][a-z]{2}_\d+)	Idd_1111
([A-Z][A-Z]-\d\d\d\d)	ID-4321

Numeric Data Pattern Examples

You want to generate credit card numbers, phone numbers, Social Insurance numbers, and Social Security numbers from regular expressions.

To generate a 10-digit phone number, consider the following scenarios:

- A phone number that starts with 3 or 4, you enter the data pattern as (3|4)\d{9}.
- A phone number that starts with 6 and consists of digits such as 2,4,6, and 8, you enter the data pattern as 6(2|4|6|8){9}.
- A phone number that starts with 6 and consists of digits between 3-7, you enter the data pattern 6 [3-7]{9}.

The following table shows the sample output for the patterns that you can enter:

Example	Pattern
Credit card number	You enter the pattern as 5\d{15} to generate a sample output as 5312003654360876.
Phone number	You enter the pattern as \d{10} to generate a sample output as 6175551212.
Social Insurance number	You enter the pattern as \d{9} to generate a sample output as 289645236.
Social Security number	You enter the pattern as \d{9} to generate a sample output as 999999999.

Credit Card Number Generation

You can generate credit card numbers based on the type of the card. You can choose from multiple credit card number formats.

You can select the credit card issuer type that you want to generate, specify the issuer identification number for the card, and the distribution percentage. You can generate test data for the following credit card types: American Express, Discover, JCB, Mastercard, and Visa. If you enter incorrect values for the credit card identification number, the workflow generation fails.

Use a credit card generation rule to generate data with string data type. You can generate formatted values for a credit card. For example, if the target is a Mastercard, you can configure the rule to generate a credit

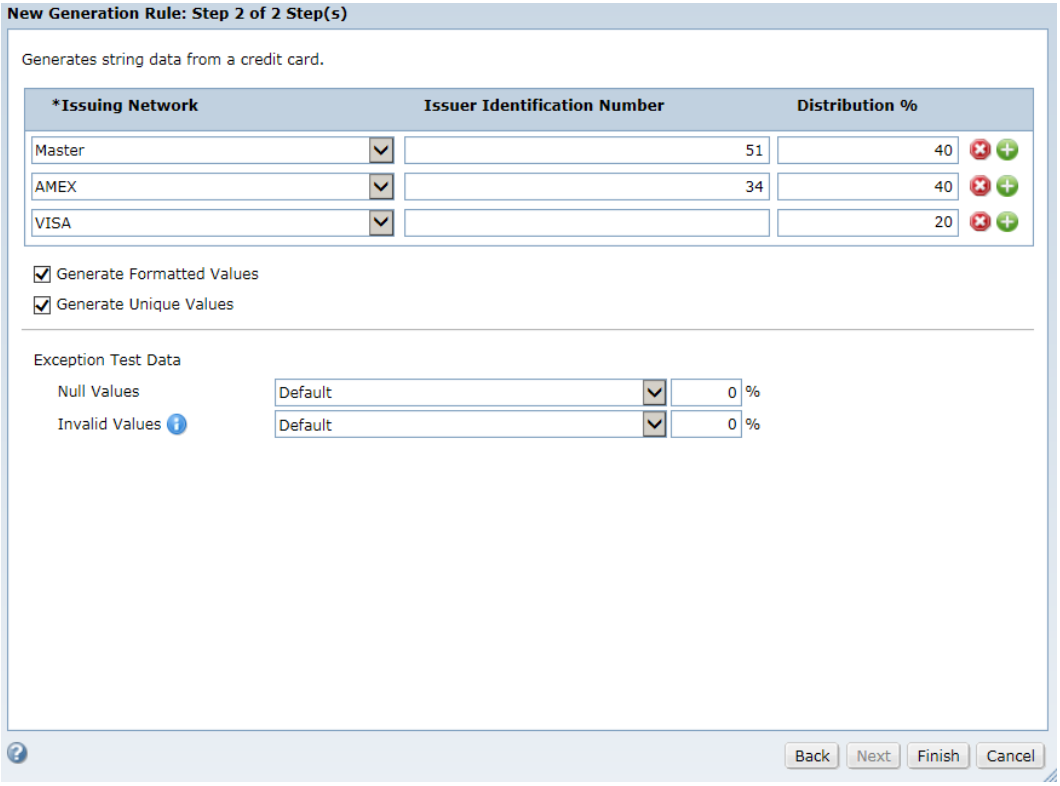
card number with the proper format of a Mastercard. The target credit card number contains numbers and hyphens in the following format:

```
XXXX-XXXX-XXXX-XXXX
```

You can generate unique values of credit card numbers. The generated credit card numbers must pass the credit card verification rules. Use Luhn's algorithm to validate the credit card number.

You cannot generate credit card numbers for numeric, date, and binary data types.

The following image shows the credit card generation parameters:



Issuer Identification Number

An issuer identification number contains the starting digits of a credit card number.

The issuer identification number accepts up to six digits. After the valid values, TDM adds any number from 0 to 9. For example, the issuer identification number for a Visa card is 4. So, TDM can generate data such as 4X, 4XX, 4XXX, 4XXXX, 4XXXXX, and so on.

If you do not specify a number, TDM generates data based on the valid issuer identification numbers for the corresponding issuing networks. If you provide a valid value, TDM generates data based on the issuer identification number. If you provide a value that is not valid, the workflow fails to run.

The following table shows the valid issuer identification numbers for different issuing networks:

Issuing Network	Issuer Identification Number
American Express	34 and 37
Discover	6011, 622126 to 622925, 644-649, and 65
JCB	3528-3589
Mastercard	51-55
Visa	4

Credit Card Generation Parameters

The following table describes the parameters that you can configure for credit card generation:

Parameter	Description
Issuing Network	Required. The type of credit card issuing network. You can choose to generate the following type of credit cards: American Express, Discover, JCB, Mastercard, and Visa.
Issuer Identification Number	Optional. The starting valid numbers for the credit card type. You can enter up to six digits. You must enter a valid value for each credit card type. If the issuer identification number is not valid for the selected credit card type, the workflow generation fails. If you leave the field blank, TDM generates data based on the valid issuer identification numbers for the corresponding issuing networks.
Distribution %	The percentage of rows that you want to generate for the specified credit card type. The sum of the distribution of the values have to be 100%. You can enter a decimal value in the distribution percent field.
Generate Formatted Values	Generates formatted values for the credit card type. For example, you can generate 16 digits for a Visa card that contains every four digits separated by hyphens.
Generate Unique Values	Generates unique data values.
Null Values	A list of global rules or default rule that are available for the string data type. A null value is a value that a default rule or a global rule generates in the target columns for the string data type.
Invalid Values	A list of global rules or default rule that you can select for the string data type. Any value that is less than the minimum length or more than maximum length for a specified range is not valid. A string that does not match the specified pattern is not valid.

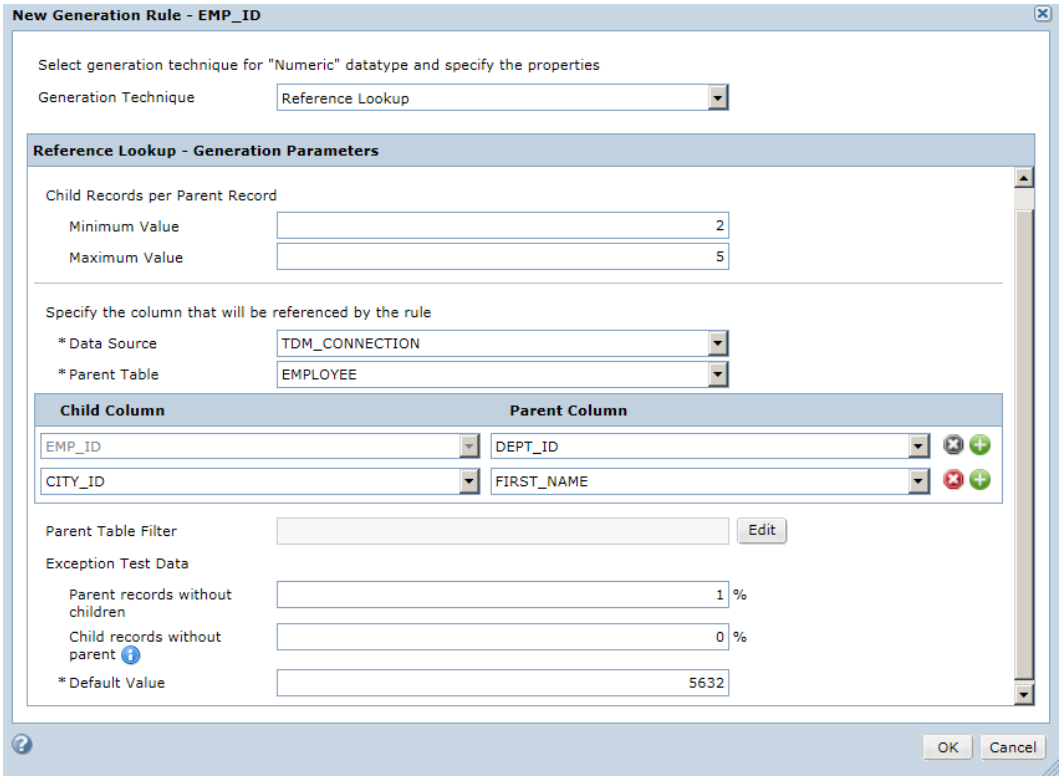
Reference Lookup Generation

A reference lookup generation rule is an ad hoc rule that generates data from a reference table. You can create reference lookup rules with the string, numeric, and date data types.

Create a reference lookup generation rule to generate column data using data from an existing table. Specify the parent table and assign the parent columns to child columns in the target database.

You must assign a parent table and parent column to the target column that you create the rule for. When you create the rule you can also assign parent columns from the same parent table to other target child columns.

The following image shows the reference lookup generation parameters:



Reference Lookup Generation Example

You want to generate customer identification numbers in a column. You want to use the same customer identification numbers that exist in a different table. You create a reference lookup generation rule that references the column in the other table and generates the values from that table.

Reference Lookup Generation Parameters

Create reference lookup rules for string, numeric, and date data types.

The following table describes the parameters that you can configure to generate data from a reference lookup table:

Parameter	Description
Minimum Value	The minimum number of child records for a parent record. Default is 1.
Maximum Value	The maximum number of child records for a parent record. Default is 3.
Data Source	The connection to the parent table.
Parent Table	The parent table that you can get the data from.
Child Column	The column name on which you create the rule. You cannot remove the first child column. To enter additional child columns, you can add more rows.
Parent Column	The parent column for the corresponding child column. You add more rows to enter additional child and parent columns. You can also assign parent columns from the same parent table to the other target child columns.
Parent Table Filter	A conditional expression that you enter to extract the data. The data you extract from the parent table must conform to this condition.
Parent records without children	The percent of parent records without a child.
Child records without a parent	The percent of child records without a parent. You can enter a value greater than 0. If you enter 0, you must provide a default value instead of null values.
Default Value	The default value that you want to generate in the target column instead of null values. The default value can be one of the values present in the parent table if the tables are in a parent-child relationship. Required if the percent of child records without a parent is set to 0. Optional if the percent of child records without a parent field is blank. Disabled if you enter a non-zero value in the percent of child records without parent field to generate null values.

Sequence Generation

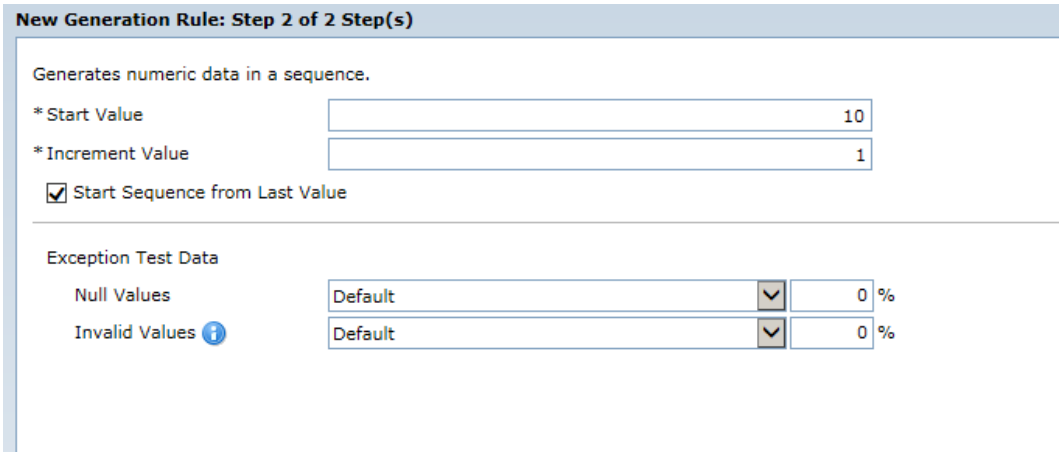
A sequence generation rule is a global or ad hoc rule that generates numeric and date values in a sequence.

You select a beginning value and an increment value when you create a sequence generation rule.

When you create a sequence generation rule for a numeric data type, you can choose to start the sequence from the last output value. When you run the plan again, the sequence continues to generate numbers from the previous output value

If you try to generate numbers in a column with string data type and if you choose to start the sequence from the last value, TDM generates values from the beginning when you run the plan again.

The following image shows the sequence generation parameters:



Sequence Generation Example

You want to generate ticket numbers in ascending order. The target table has a TICKET_NUMBER column. You create a sequence generation rule with the numeric data type. You enter a start value of 1000 and an increment value of one (1).

Date Sequence Generation Parameters

Create sequence generation rule for date data types.

The following table describes the parameters that you can configure to generate a sequence of data for date data type:

Parameter	Description
Start Date	The date at which you can start generation of sequence data for date data types.
Increment by	The value to increment after the start date. You can enter a number and select year, month, day, hour, minutes, or seconds.
Null Values	A list of global rules or default rule that are available for the date data type. A null value is a value that a default rule or a global rule generates in the target column for the date data types.
Invalid Values	A list of global rules or default rule that are available for the date data type. Any value that is less than the sequence start value for a positive increment or more than the sequence start value for a negative increment is not valid.

Numeric Sequence Generation Parameters

Create sequence generation rule for numeric data types.

The following table describes the parameters that you can configure to generate a sequence of data for numeric data type:

Parameter	Description
Start Value	The minimum value to start generation of a sequence. You can enter a number to generate the sequence data.
Increment Value	The value to increment after the start value during generation of the sequence data. You can enter a number to increment the value.
Start Sequence from Last Value	Starts the sequence from the last output value when you run the plan again. **Note:** If you try to generate numbers in a column with string data type and if you choose to start the sequence from the last value, TDM generates values from the beginning when you run the plan again.
Null Values	A list of global rules or default rule that are available for the numeric data type. A null value is value that a default rule or a global rule generates in the target column for the numeric data type.
Invalid Values	A list of global rules or default rule that are available for the numeric data type. Any value that is less than the sequence start value for a positive increment or more than the sequence start value for a negative increment is not valid.

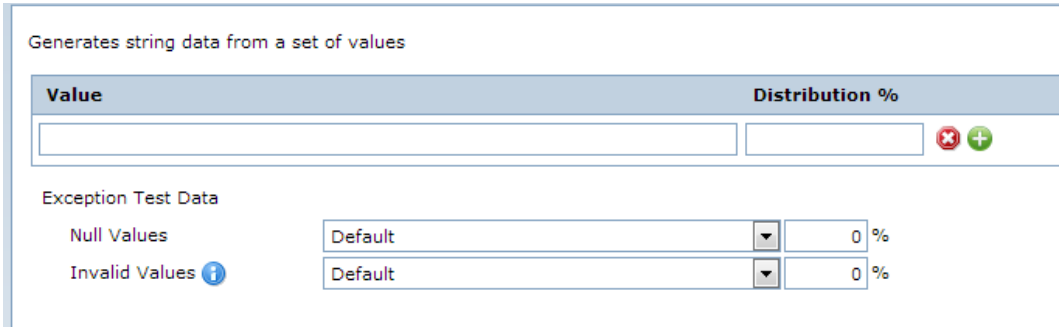
Set of Values Generation

A set of values generation rule is a global or ad hoc rule that writes a small data set to the target table. You can list the data values that you want to write to the table when you create a set of values generation rule.

You can use the string, numeric, and date data types in a set of values generation rule. You add data values in the rule and specify what percent of the rows in the target table you want to have the value. You can enter the percent in decimal values. You can use set of values generation for columns that have a small number of values, such as membership status.

Set of values rules can be global rules or ad hoc rules.

The following image shows the set of values parameters:



Set of Values Generation Example

You want to generate payment information. The target database has a PAYMENT_TYPE column with Visa, MasterCard, and Paypal values. You want 60% of the rows to have the Visa value, 30% of the rows to have the MasterCard value, and 10% of the rows to have the Paypal value.
You create a set of values rule with the string data type. You add Visa in the value field and enter 60 in the distribution percent field. You click the add button twice to add two additional values to the rule. In the second row, you enter MasterCard in the value field and 30 in the distribution percent field. In the third row, you enter Paypal in the value field and 10 in the distribution field.

Set of Values Generation Parameters

Configure set of values parameters to generate data for string, date, and numeric data types.

The following table describes the parameters that you can configure to generate a set of values:

Parameter	Description
Value	The exact data value that you want to write to a target table. You can enter exact values for the string and numeric data types. You enter a date for date data type. If you want to use more than one value, you can add more rows.
Distribution%	The percent of rows for the specified value. The sum of the distribution of the values have to be 100%. You can enter a decimal value in the distribution percent field.
Null Values	A list of global rules or default rule that are available for the data type you select. A null value is a value that a default rule or a global rule generates in the target columns based on the data type you select.
Invalid Values	A list of global rules or default rule that are available for the data type you select. Any number, date, or string that is not a part of the set of values is not valid.

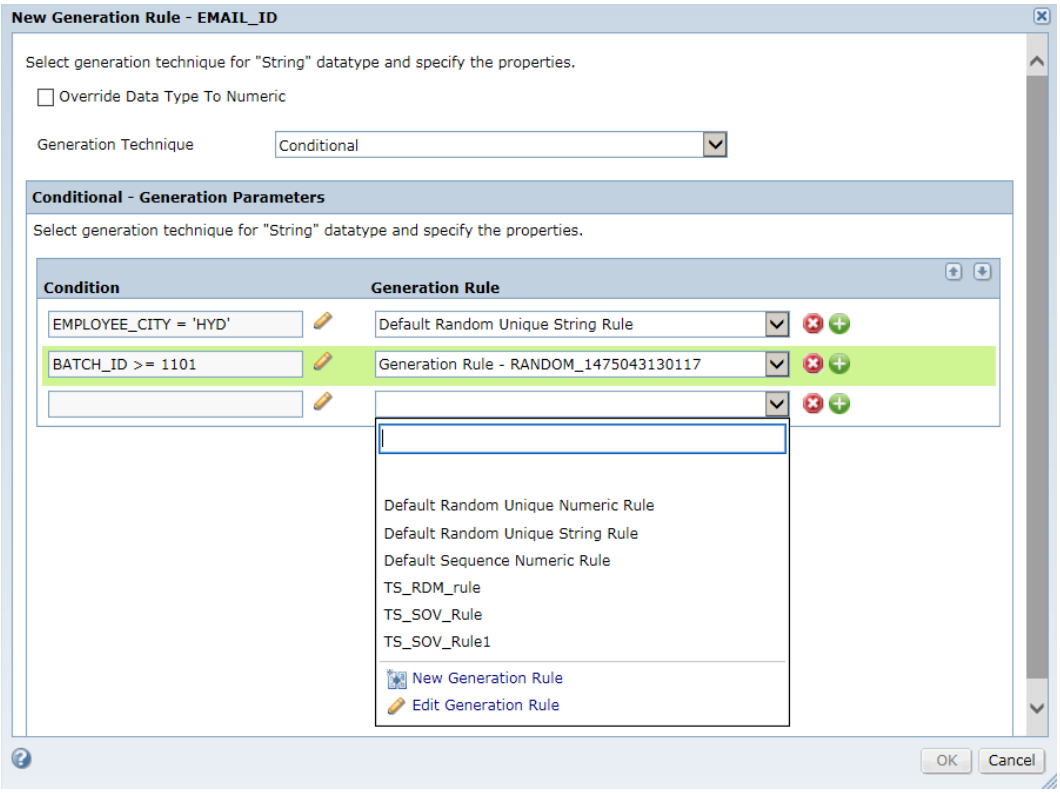
Conditional Generation

A conditional generation rule is an ad hoc rule in which you can specify a conditional expression and a generation rule to generate test data.

You can use the string, numeric, and date data types in a conditional generation rule. You can apply multiple conditions for a generation rule. When you run a data generation plan, TDM applies the conditions based on the order specified. You can use the up and down arrows to change the order of the conditions.

You cannot apply conditions on effective dates and sequence generation rules.

The following image shows the conditional generation parameters:



Conditional Generation Example

You work for the Human Resources department and you need to decide on the hike and bonus percentages for employees based on the departments. If the DEPT_ID is 11, give a hike of 5%. If the DEPT_ID is 22, give a hike of 10%. If the DEPT_ID is 33, give a hike of 15%.

You can apply a conditional ad hoc generation rule on the HIKE column. You can specify the conditions for each department and select a generation rule. The DEPT_ID column must have a generation rule assignment. When you run the plan, TDM generates data for the DEPT_ID column first. Based on the DEPT_ID column, TDM generates data in the HIKE column.

You can also assign bonus percentages with the HIKE column as the dependent column. Assign a conditional generation rule to the BONUS column and add conditions to the HIKE columns to assign bonus percentages.

Conditional Generation Parameters

The following table describes the parameters that you can configure for conditional ad hoc generation:

Parameter	Description
Condition	A conditional expression based on which you can generate test data in a target column. You can add multiple conditions for a column. Use Expression Builder to create conditions.
Generation Rule	The generation rule that you can apply to the target column based on the specified condition. You can use the existing generation rules in the project, or you can create a generation rule. You can also edit an ad hoc generation rule.

CHAPTER 11

Working with Test Data Warehouse

This chapter includes the following topics:

- Test Data Warehouse Overview, 227
- Test Data Warehouse Process, 228
- Data Sets, 229
- Test Data Management Self-Service Portal, 230
- Creating a Data Set, 230
- Reset a Data Set, 231
- Editing the Metadata of a Data Set, 238
- Publishing a Data Set to the Self-Service Portal, 238
- Deleting a Data Set, 239
- Related Data Sets, 239
- Data Set Permissions, 243
- Locking and Unlocking a Data Set, 245
- Monitor a Data Set Job, 245
- View and Manage Data in a Data Set, 246

Test Data Warehouse Overview

Create a test data warehouse to create a central location that stores multiple versions of test data. Multiple Test Data Manager services can access the same test data warehouse to create a central storage location across teams.

To create a test data warehouse, you create a Test Data Warehouse Service from the Administrator tool. The Test Data Warehouse Service administers the test data warehouse repository and the test data warehouse.

Create and manage data in the test data warehouse from Test Data Manager. Data is stored as data sets in the test data warehouse. The test data warehouse repository stores the project metadata and the source table metadata associated with the data set . Project metadata includes the plan information, version, and tag information. The test data warehouse stores the source data that you include in the data set.

You can create multiple versions of a data set to maintain different versions of the test data. To return a test environment to a previous state, you can choose to restore a specific version of the data from the test data warehouse.

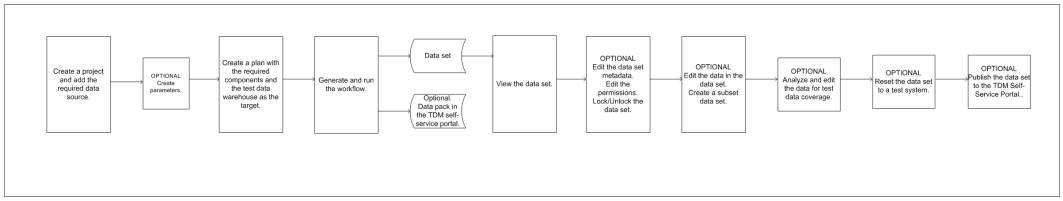
You can publish a data set that you create in Test Data Manager to the Test Data Management Self-Service Portal. Testers who do not have access to Test Data Manager can view and restore required test data to a test environment.

For example, before you run any test cases, you create a data set to store the original test data in the test data warehouse. After each team runs its tests and the data is modified, each team creates another version of the data set to store the modified test data. To return the test environment to its original state, you can restore the version of the data set that contains the original data. You can retrieve any of the other data set versions to return a test environment to a required state.

Test Data Warehouse Process

Store data in the test data warehouse to manage multiple versions of test data in a central location.

The following image shows the tasks that you perform to store data in the test data warehouse and the tasks that you can perform on the data in the test data warehouse:



You can perform the following tasks on data that you store in the test data warehouse:

Edit metadata

You can edit the tags and the description that you add to a data set.

Change permission settings

You can change user permission levels for a data set to control the level of access to the data set.

Lock and unlock a data set

The administrator and the data set owner can lock and unlock a data set. Lock a data set to prevent users from editing or resetting a data set to a target test system. For example, if you want to check the data before users reset a system with the data. A data set owner can perform a reset on a data set that is locked.

Reset

You can reset a data set to a target system to return the test data on the system to a specific state.

Edit the data

You can edit the data in a data set. You can update the data, add or delete rows, and add row-level tags to the data.

Create a related data set

You can create a subset of a data set based on required criteria.

Analyze the data for test data coverage

You can analyze the data in a data set based on data requirements for test cases and then edit the data for better test case coverage.

Delete

You can delete a data set from the test data warehouse.

Data Sets

A data set is a collection of data that consists of one or more tables from a data source. Manage and store multiple versions of data sets in the test data warehouse.

Tables in a data set can be from a single data source or from multiple data sources. A data set generally corresponds to one or more test cases. You can create data sets based on the test cases that you want to run, and store them in the test data warehouse. You can create a data set from TDM when you perform one or more of the following tasks:

- Create a subset of production data.
- Create a masked copy of production data.
- Generate data.

To create a data set, select the test data warehouse as the target when you run a plan. You can add tags and other metadata information to identify the data set. Metadata includes the source table metadata and plan metadata. You can search the test data warehouse based on the metadata properties.

Test data might change when you use the data to run test cases. You can store the changed data as another version of a data set in the test data warehouse. You can create multiple versions of a data set.

You can create another version of a data set in the following ways:

- Rerun the plan that you used to create the data set without changing the name of the data set.
- Create and run a different plan and enter the same data set name.

Different versions of a data set can have similar metadata or different metadata. For example, you create a data set version DA_APP when you test version 1.0 of an application for a specific customer. You test specific features of the application in this release. You use source data that contains tables relevant to the tests you run. You might add tags to identify the customer. When you test the next version of the application for the customer, you test different features that require different test data. You create and run a different plan with a different data source. Because this is test data for a single application, you enter the same data set name to create another version of the data set. The version retains the tags that identify the customer. This version contains different data and metadata.

Test data on a test environment can get corrupted. You might require the original test data to run additional tests. You can return a test environment to a specific previous state when you reset a specific data set from the test data warehouse to the required connection.

For example, consider a situation where test teams run multiple test cases, or when multiple test teams work on an application. Store the original test data as a data set in the test data warehouse. When one test team completes testing, create another version of the data set with the modified test data. You can then reset the original test data from the test data warehouse to the test environment. You can reset any version of a data set to return the test environment to the required state.

View the list of data sets in the **Data Sets** view.

Data Set Tags

You can add tags to a data set to classify and identify each version of the data set. You can also add row-level tags to a data set. Use a comma to separate the tags.

A data set contains test data that you require to run specific test cases. You can add the name of the test case, an identification number, or any relevant information as a tag to identify the data set. You can perform a keyword search for data sets based on tags. When multiple test teams create data sets, the teams can add tags to identify the data sets. Share tag information to help users search for a data set that you want to share.

Use tags to filter and search for data sets in the **Data Sets** view.

Use row-level tags to identify data when you edit data in a data set or as criteria to create a related data set.

Test Data Management Self-Service Portal

You can publish test data that you save in the test data warehouse to the Test Data Management Self-Service Portal.

Users who do not have access to Test Data Manager use the self-service portal to access test data. You can view and manage test data and copy required test data to a test environment from the self-service portal.

You can publish test data as data packs to the self-service portal. You can choose to publish the test data to the self-service portal from the plan that you run to create a data set in the test data warehouse. You can also choose to publish existing data sets to the self-service portal.

Creating a Data Set

Run a plan to create a data set. Create and run different plans for data generation operations.

- You cannot create a data set if the tables contain columns with user-defined data types. If a table contains columns with user-defined data types, mark the columns as restricted before you include the tables in a plan to create a data set. To mark a column as restricted, open the project and click **Discover** > **Columns** and select the option **Restricted** for the required column.
- Verify whether the tables in the data source contain primary keys. Create logical keys in all transactional tables that do not have any primary keys. A reset operation might fail if a transactional table with null values does not have primary keys.
- If a table contains more than two character columns with a precision of 4000 each, edit the precision to 3,999 or less. A reset operation might fail if a data set contains more than two character columns with a precision of 4,000 each and you do not disable constraints during the reset operation.
- You cannot perform self-service portal tasks that require a staging connection if a table in the data pack contains columns with a precision of 4,000 or greater. Edit the precision to 3,999 or less.

1. Open a project and add the required data sources and the project components.
2. Optional. Create parameters to use in the plan.
3. To create a plan, click **Execute** and then click **Actions** > **New**.
4. Enter a name and optional description for the plan. Click **Next**.

5. Add and configure the required components.
6. Configure the plan settings.
7. To create a data set, select **Test Data Warehouse** from the list of target connections.

   The **Data Set Properties** section appears.
8. Enter a data set name. TDM enters the plan name by default. You can edit the name.

   When you enter a data set name that does not exist in the test data warehouse, you create the first version of a data set. TDM enters the data set version number.
9. Enter an application name and application version to identify the data set.
10. Add tags to the data set. Use a comma to separate tags.

    To delete a tag that you enter, click the **X** icon that appears with the tag. To use the keyboard to delete a tag, press **Tab** or **Shift+Tab** to select the tag and then press **Delete** from the keyboard.
11. Configure the error and recovery settings and advanced settings as required.
12. Optionally, configure an override strategy for a data source or a table.
13. To save the plan, click **Finish**.
14. To generate and run the workflow, click **Actions** > **Generate and Execute** in the **Plan Properties** page.
15. Optional. To publish the test data to the self-service portal, enter a name and select users or user groups to assign permission. You can add tags and enter a description for the data pack.

You can view the job log messages in the **Monitor** page. When the job completes successfully, you can view the data set in the **Data Sets** view.

Related Topics:

- "Project Components" on page 49
- "Creating a Project" on page 52
- "Executing a Workflow" on page 295

Reset a Data Set

Reset is the process of moving data from a data set in the test data warehouse to a target.

During the course of testing, the test data might change. You might want to reset the data in a test environment with a version of the data set from the test data warehouse.

For example, you are responsible for testing certain features of an application. You create test data to use in the test cases that you run. Before you use the data, you save the data as a data set in the test data warehouse. When you complete tests on some features, the data might change or get corrupted. You might require the original test data to run the rest of the test cases. You can replace the data in the test system with the data set in the test data warehouse. Perform a reset operation from the data set to a target. Use the test system as the target.

You can configure a reset operation to delete the target data and replace it with the data in the data set. In this case, the reset operation restores the target data to the exact same state as the data set. Perform this reset operation if testers do not share the tables. You do not need records that exist in the target and not in the data set.

You might want to retain data in the target that is not in the data set. Consider a case where multiple testers use the test data. You need the first 100 records in a Customer table. You create a data set that contains the

first 100 records of the Customer table. After running a few tests, you want to replace the test data with the data set. You must configure the reset operation to ensure that the additional data in the target remains untouched when you reset the data in the data set.

In reset operations that selectively delete and load data, TDM runs an SQL script to create a backup of the deleted data in the target.

You can select the required configuration when you configure the reset operation.

Classification of Data Set Tables

You can classify the tables in a data set as master tables or transactional tables. The results of the reset operation depend on how you classify the tables.

A reset operation does not delete records in master tables. It performs an upsert on master tables. When testers share all records in a table, classify the table as a master table. Because the reset does not delete records in master tables, the reset operation does not delete records that might impact the work of other testers.

Classify tables that are descendants of master tables as transactional tables. Records in transactional tables are generally not shared records. When you delete a record, you do not impact the work of another tester. By default, during a reset operation TDM performs an insert on transactional tables. You can choose to change the default behavior when you configure the reset operation.

If you do not classify a table, by default a reset operation considers the table as a transactional table.

TDM performs the following tasks during an upsert:

- Updates records that exist in both the data set and the target.
- Inserts records that are present in the data set and not in the target.

During an insert, TDM inserts records from the data set that do not exist in the target. If the record exists, the insert fails.

Based on how you classify tables as master and transactional, TDM identifies top-level transactional tables and begins the cascade delete at this level.

A top-level transactional table is a direct child of a master table. All parents of a top-level transactional table, if any, are master tables.

Verify the table classification before you perform a reset operation.

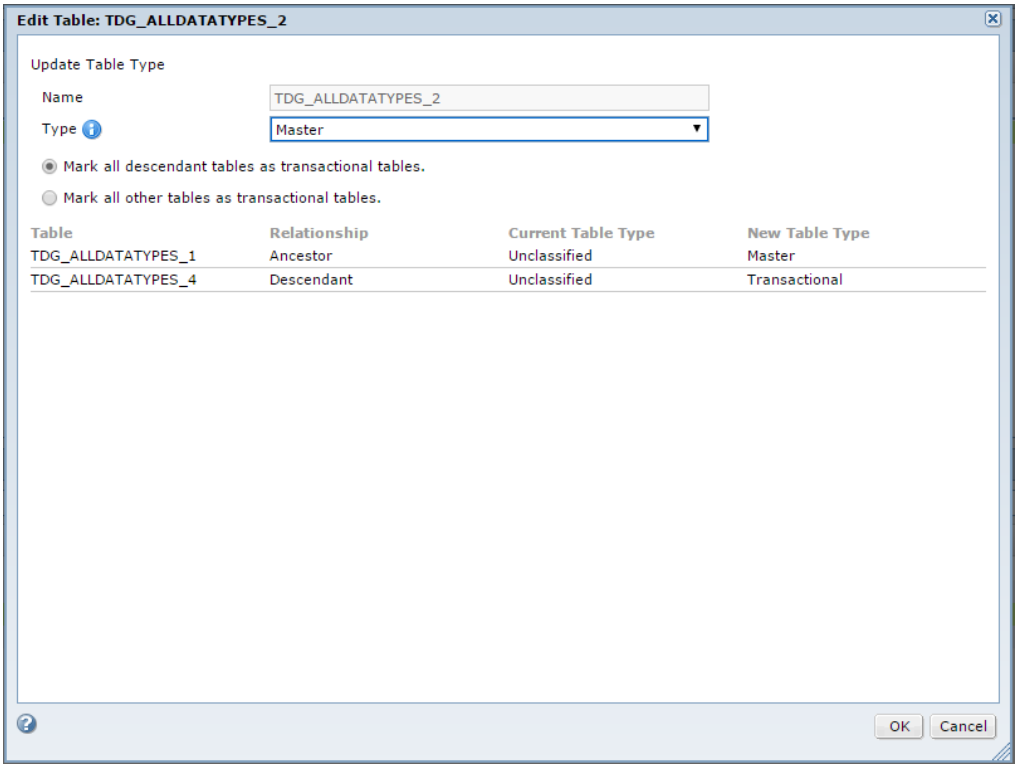
Classifying a Data Set Table

Classify tables in a data set as master tables or transactional tables to control the results of a reset operation.

1. Open the required data set.
2. Click the **Plan Components** tab.
3. Click the **Tables** tab in the panel below the list of plan components to display a list of tables and the properties.
4. Click **Edit Table Type** to open the **Edit Table** dialog box.
5. Select the required table type from the list of table types.
6. If you classify a table as a master table, you must choose one of the following options:

- Mark all descendant tables as transactional tables. TDM marks all related descendant tables as transactional tables.
- Mark all other tables as transactional. TDM marks all related descendant tables and all unrelated tables as transactional tables.

A list of the tables in the data set with the impact of the classification on the tables appears. The following image shows the **Edit Table** dialog box:



7. Click **OK**.

Reset With and Without Downtime

You can choose to disable constraints and indexes when you perform a reset operation.

Disabling constraints during the reset might increase the performance of the reset operation because the constraints do not exist. TDM performs delete and load in multiple tables in parallel and the reset operation might take less time to complete.

When you do not disable constraints between tables during a reset operation, TDM must check the order in which deletes and inserts happen. TDM cannot perform load and delete for multiple tables in parallel. TDM must consider constraints before deleting records. The reset operation might take longer to complete than when you disable constraints and indexes.

You apply constraints between tables. When you disable constraints, you disable constraints for the entire table. During the time the constraints are disabled, tests that you run using the tables might fail as the constraints do not exist. Consider a case where multiple testers use the test data. You create a data set that contains the first 100 records of the Customer table. After running a few tests, you want to reset the test data with the data set. If you disable constraints for the table during the reset operation, tests run by other testers that include the Customer table might fail. The reset operation causes a downtime for running tests that use the target connection.

While disabling constraints can increase performance, there might be a tradeoff with a downtime in testing if the target tables are shared among testers.

If multiple testers use the tables, a slower reset operation might be better than a downtime in testing for multiple testers. If the target tables are not used by other testers, and you are sure that no tests that include the tables are run during the reset, you can increase the performance of the reset by disabling constraints.

Analyze the requirement and consider this tradeoff when you configure a reset operation.

Resetting a Data Set

You can reset a specific version of a data set to a target. Reset a data set to a target to reuse a specific version of test data or to reset test data to an earlier state. You can configure the reset operation based on how you want to replace the test data.

As a best practice, create an index on foreign keys on the target database. During the deletion, the index helps find the child records in the target database. The reset performance improves because the deletion of child records takes less time.

Ensure that you classify the tables in the data set before you perform a reset operation. TDM considers unclassified tables as transactional tables during a reset operation.

1. Open the required version of the data set.
2. Click **Actions** > **Reset**.

   The **Reset** dialog box opens.
3. Select the target connection from the list of connections.

   The source is the test data warehouse. You cannot edit the source. Select the database where you want to replace the data as the target connection.
4. Configure the settings for the reset operation:
   - Truncate Tables. Select this option if you want to completely replace the target connection with data from the data set. TDM deletes all data in the target and copies the data in the data set to the target. TDM disables constraints and indexes when it performs this operation.
   - Disable Constraints and Indexes. Select this option if you want to disable constraints and indexes during the reset operation.
   - Upsert in Transactional Tables. Select this option if you want to perform an upsert on transactional tables instead of an insert. If the parent records of records in the data set have changed to point to a different parent in the target, the insert might fail. The reset operation fails if the insert fails. Select this option and perform an upsert on transactional tables.
   - Insert in Master Tables. Select this option to perform an insert on master tables instead of an upsert. Select this option if you are sure that no records from the data set exist in the target. Instead of first checking if any records exist and need to be updated, the reset directly inserts all records from the data set into the target. To use this option, all tables in the data set must be classified as master tables.
5. Enter the reset recovery properties.
   - Backup Recovery Prefix. Enter a prefix for the backup schema. TDM creates a backup schema in the format *<prefix>_<tablename>*.
   - Backup Recovery User Name. The name of the schema where you want to store the backup tables.

   You cannot enter the reset recovery properties if you choose to truncate tables. TDM does not create a backup when it truncates tables.
6. Edit the **Error and Recovery** and **Advanced Settings** as required.

7. Click **Next**.
8. Optional. You can override the reset settings at the data source level or at the table level.
9. Select the PowerCenter Integration Service to use for the reset operation. You must select the service that you configured in the Test Data Manager Service.
10. Select to run the operation or schedule it to run later.
11. Click **OK**.

You can view the job log messages in the **Monitor** tab.

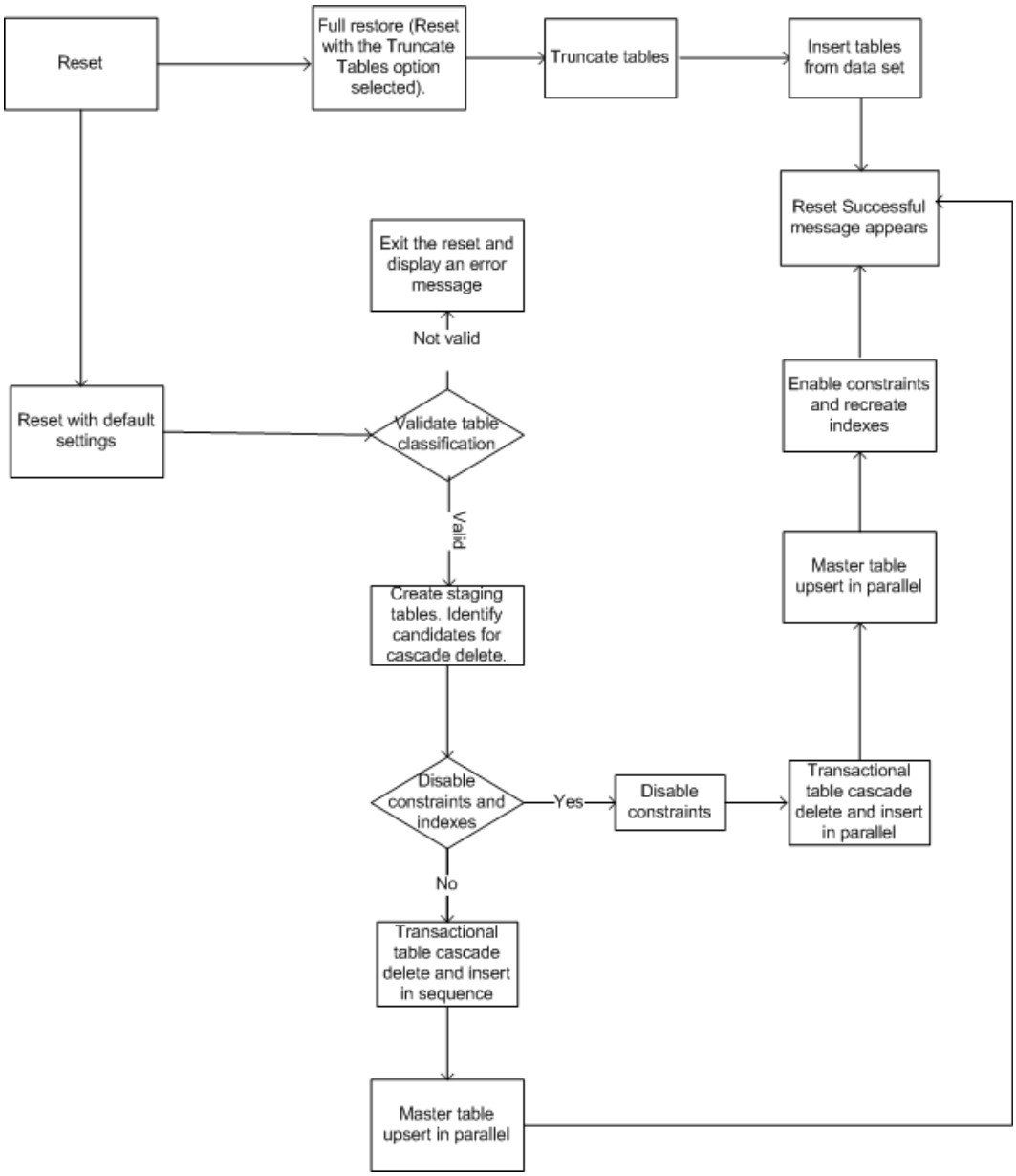
Reset Process Flow

You can perform a reset operation that truncates tables or a reset operation that does not truncate tables.

Perform a reset with the truncate tables option when you own the target schema or when you do not share the data. By default, TDM disables the constraints and indexes during the reset. The downtime that this causes does not impact the work of other users because you do not share the tables.

Do not truncate tables if you share the table data and you need to reset a part of the data. You might delete data that other users need.

The following image indicates the steps that a reset operation goes through:



Note: The image does not include tasks performed during backup.

Reset with Truncate Tables Process Flow

You can choose to truncate the tables when you perform a reset operation.

TDM performs the following tasks when you perform a reset operation and select the **Truncate Tables** option:

1. Disables the constraints and indexes.
2. Deletes all tables in the target.
3. Inserts all tables from the data set to the target.

4. Enables the constraints and indexes.

Reset Without Truncate Tables Process Flow

You can perform a reset operation without truncating the tables.

TDM performs the following tasks when you perform a reset operation and do not select the truncate tables option:

1. Identifies records for deletion from transactional tables that have child tables.
2. If there are no records to delete, TDM skips this step. If records for deletion exist, TDM performs the following tasks.
   a. Creates staging tables for the identified records and stages the keys.
   b. Identifies top-most transactional tables in the data set.
   c. For each record in a top-most transactional table in the target, if a record exists in the data set, TDM marks the record in the staging table.
   d. Runs an SQL script to cascade mark records in the staged tables for deletion. If a parent record is marked for deletion, TDM marks the child record for deletion.
   e. Checks if recovery is enabled for the reset operation and that the user name and prefix information are available.
   f. If the recovery option is enabled and the prefix information provided, TDM performs the following tasks. If the recovery option is not enabled, TDM skips this step.
      a. Creates temporary tables in the backup schema and then moves records from the staging area to the backup schema.
      b. Stores a backup of all records marked for deletion in the backup tables in the backup schema. TDM runs an SQL script to back up the tables and uses the naming convention *<prefix>_<targetname>*.
3. Checks if the **Disable Constraints and Indexes** option is enabled. If the option is enabled, the reset operation performs the following steps:
   a. Disables the constraints and indexes.
   b. Deletes marked records from the target. TDM reads the marked records from the staging tables.
   c. Moves data from the data set to the target. TDM processes multiple tables in parallel because the constraints and indexes are disabled.
   d. Enables the constraints and indexes.
   If the option is not enabled, the reset operation performs the following steps:
   a. Deletes marked records from the target. TDM reads the marked records from the staging tables.
   b. Moves data from the data set to the target. TDM processes the tables according to constraints because the constraints exist.
4. Deletes the temporary tables, if created.
5. Deletes the staging tables, if created.
6. If the reset operation is successful, TDM deletes the backup tables. If the reset operation is not successful, TDM retains the backup tables and displays an error message.

Restoring Deleted Data After a Failed Reset

If you need to restore data that was deleted during a failed reset operation, you can restore the data from the backup created during the reset. Run a separate plan to restore the data from the backup. Ensure that you have the backup prefix used during the failed reset operation.

Perform the following high-level steps to restore data from the backup schema:

1. Create a group that contains all the tables that were backed up during the reset operation.

   The backed up tables include all transactional tables.
2. Open a project and create a plan.
3. Add the group as a subset component to the plan.
4. Add the connection that contains the backup schema as the source connection.
5. Add the database where you want to restore the data as the target connection.
6. In the target settings in the plan, disable the constraints and indexes.

   Select the following values for the **Update Strategy** properties:

   - **Treat Source Row As**: Update.
   - **Update As**: Update else Insert.
7. In the **Advanced Settings**, enter the backup prefix in the **Reset Recovery Prefix** field.
8. Run the plan.

Editing the Metadata of a Data Set

You can edit the description and tags of a data set.

1. To open the data set, click the required version of the data set in the **Data Sets** view.

   The data set page opens on the **Overview** page.
2. To edit the description or tags, click **Edit** on the **General** tab.
3. Add or edit the description as required. You can enter a maximum of 4,000 characters.
4. To add a tag, enter the text in the **Tags** field. You can use special characters and spaces in a tag. Use a comma to separate tags.
5. Click **Save**.

Publishing a Data Set to the Self-Service Portal

You can choose to publish a data set to the self-service portal. Test Data Manager publishes a data set to the self-service portal as a data pack.

1. To open the data set, click the required version of the data set in the **Data Sets** view.

   The data set page opens.
2. To publish the data set to the self-service portal, click **Actions** > **Publish to the Self-Service Portal**.

   The **Self-Service Portal** dialog box appears.

3. Enter a name and optional description for the data pack.
4. Optional. Add tags to the data pack. Use a comma to separate tags.
5. Optional. To add users or user groups for the data pack, click the **Select Users** button.

   The **Select Users for the Data Pack** dialog box appears.
6. Optionally, you can filter and search for users or user groups.
7. Optional. Select the users or user groups to add to the data pack and click **Select**.

   Note: TDM adds the current user and administrator users to the data pack by default. You can add other users if required.
8. Click **OK**.

   Note: If you choose to publish a data set that has already been published to the self-service portal, you can update the description, tags, and users or user groups. You cannot change the name of the data pack or publish the data set to the self-service portal again.

Deleting a Data Set

You can delete a data set or specific versions of a data set from the test data warehouse. Delete a data set from the **Data Sets** view. When you delete a data set, you delete the data and the metadata related to the data set.

Note: You cannot delete a data set that you publish to the self-service portal. You must delete the data pack from the self-service portal before you can delete the data set.

1. Open the **Data Sets** view.
2. You can delete a data set or specific versions of data sets.
   - To delete a data set, select the data set. TDM selects all versions of the data set. You can select multiple data sets.
   - To delete a specific version of a data set, select the version. You can select multiple versions of different data sets.
3. Click **Actions** > **Delete**.

Related Data Sets

You can create a subset of a data set to create smaller chunks of test data that are referentially intact.

A related data set is a subset of a data set. Create a related data set that contains specific data from another data set. For example, you create a data set DA_APP when you test version 1.0 of an application for a specific customer. You test specific features of the application in this release. The data set DA_APP contains tables relevant to the tests you run. When you test the next hotfix version of the application, you test a few of the features and you require some of the data in the data set.

Create a subset of the data set that includes the required data. Create an entity or group in the data set that includes the tables that you want to include in the subset. Run a plan that includes the entity or group to create the subset data set.

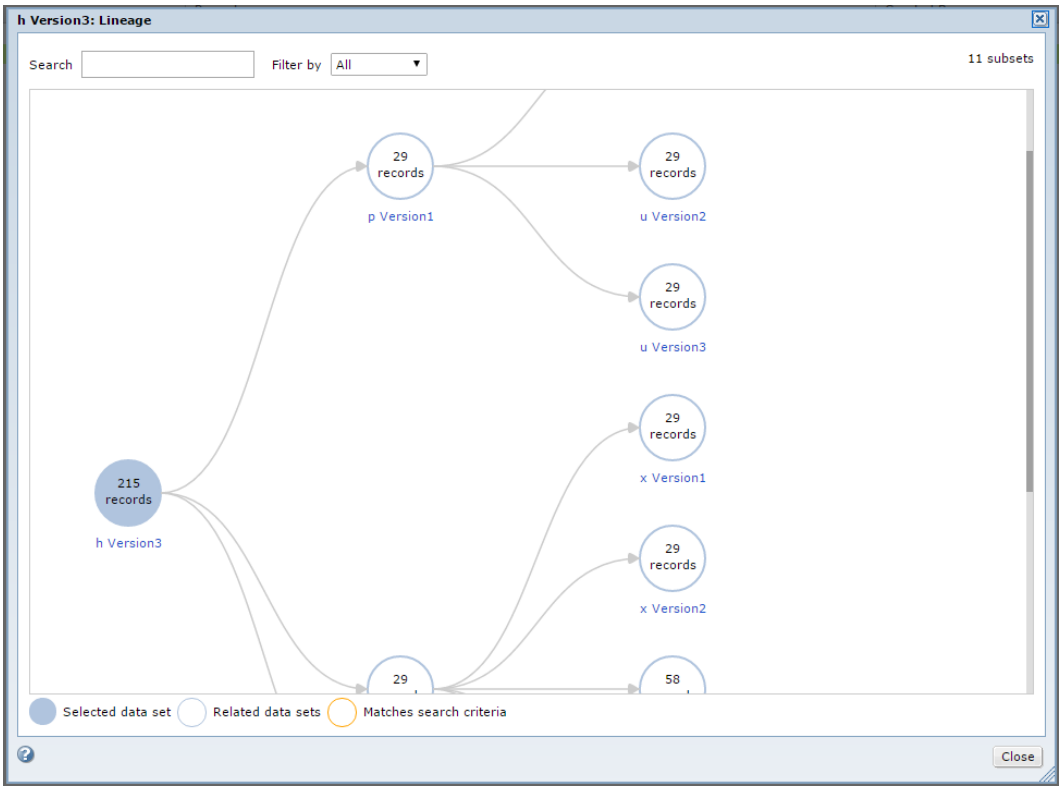
Related Data Set Lineage

You can view a map of related data sets from the **Related Data Sets** tab.

The **Lineage** window displays a map of related data sets and indicates which data sets have subsets.

You can search for and view the lineage based on specific search criteria that you enter. Results that match are highlighted. You can filter the view to view all parent data sets, all child data sets, or all related data sets.

Click the name of a data set to open the data set page. You can open data sets that you have permissions on.



Creating an Entity

You can create an entity with tables in a data set. When you create an entity, you select the driving table. Test Data Manager retrieves the related tables in the data set based on the constraints. You can add filter criteria to the driving table to filter source data when you create the subset of the data set. You can disable relationships between tables in the entity.

1. Open the required data set.
2. Click **Related Data Sets** and open the **Subset Components** tab.
3. Click **Actions** > **New** > **Entities**.
4. In the **New Entity** dialog box, enter a name and optional description for the entity.
5. Click **Select Table** to select a driving table from the list.
6. Click **Save**.

   Test Data Manager adds the related tables to create the entity and displays a diagram of the relationship between the tables in the entity. You can view a list of the tables or a list that shows the relationships between the tables.

7. To create a subset of the data based on filter criteria, click **Edit** in the **Properties** pane.

   The **Edit Criteria** dialog box appears.
8. To add criteria for a column, click the **Entity Criteria** tab.
9. Click the **Add Criteria** button.
10. Select the columns on which you want to filter data and click **OK**.

    The columns that you select appear in the columns list in the **Entity Criteria** tab.
11. To define the filter expression in the entity, select the attribute from the Attribute list. Choose **Value** to enter a value or **Parameter** to enter a parameter name.

    You can indicate a filter condition in the entity, but define the expression in the plan. You can choose to create multiple criteria with parameters and include required parameters when you run a workflow.
12. Select an operator from the list to filter the data.
13. If you enter a filter condition, enter a specific value or choose the variable from the list of variables to complete the filter expression.
14. Click **Save** for each filter criteria that you create. If you define multiple filters in an entity, the filter conditions act as "AND."
15. Click **Save** to exit the **Edit Criteria** dialog box.

Editing an Entity

You can edit an entity to change the general properties, relationships, and filter criteria for tables.

1. Open the required data set.
2. Click **Related Data Sets** and open the **Subset Components** tab.
3. Open the entity that you want to edit.
4. Click **Actions** > **Edit**.

   The **Edit** dialog box appears.
5. On the **General** tab, edit the entity name and description.
6. On the **Entity Criteria** tab, select columns and enter filter criteria.
7. Click **OK**.
8. Optional. You can choose to optimize all the relationships in the entity or edit required relationships between tables.

Related Topics:

- "Entity Views" on page 108
- "Optimizing Relations" on page 113

Creating a Group

You can create a group of tables in a data set. Add the group to a plan to create a related data set. To create a group, select the tables that you want to add to the plan.

1. Open the required data set.
2. Click **Related Data Sets** and open the **Subset Components** tab.
3. Click **Actions** > **New** > **Groups**.

4. In the **New Group** dialog box, enter a name for the group and optional description for the group. Click **Next**.
5. To select one or more tables for the group, click the **Add Tables** button.
6. Optionally, you can filter the list of tables to search for.
7. Select the required tables and click **OK**.
8. Click **Finish**.

Editing a Group

You can edit a group to change its description and tables.

1. Open the required data set.
2. Click **Related Data Sets** and open the **Subset Components** tab.
3. Open the group that you want to edit.
4. Click **Actions** > **Edit**.

   The **Edit** dialog box appears.
5. On the group **General** tab, edit the group name or the description.
6. On the **Tables** tab, change the tables in the group.
7. Click **OK**.

Creating a Related Data Set

Create a related data set to create a subset of a data set that contains required data that is referentially intact.

1. Open the required data set.
2. Click the **Related Data Sets** tab.

   The **Data Sets** tab opens with a list of related data sets.
3. Click the **Execute** tab.
4. Click **Actions** > **New Plan**.
5. In the **New Plan** dialog box, enter a name and optional description for the plan.

   The plan name is the default data set name. You can edit the name of the data set later.
6. Enter an application name and application version to identify the data set.
7. Optional. Add tags to the data set. Use a comma to separate tags.
8. Click **Next**.
9. In the **Subset Components** window, click **Add** and select the required subset components to add to the plan. Click **OK**.
10. Click **Next**.
11. To filter subset components, select the component, click **Edit Table Criteria** and enter an expression. Click **OK**.
12. Click **Next**.
13. Configure the plan properties.
14. Click **Finish**.

The plan appears in the **Execute** tab.

15. To generate and run the workflow, click **Actions** > **Generate and Execute** in the **Plan Properties** page.

    You can view the job log messages in the **Overview | Monitor** page. View workflow and session logs from the **Monitor** tab on the data set page. When the job completes successfully, you can view the data set in the **Related Data Sets** tab.

Related Topics:

- "Error and Recovery Options" on page 274
- "Advanced Options" on page 275
- "Log Settings" on page 279

Copying a Data Set Plan

Copy a data set plan to create another plan with similar components. Create a copy of a plan and edit the copy.

1. Open the required data set and click **Related Data Sets** > **Execute** to view the data set plans.
2. To select a plan, click the plan **Description** or **Status** field.

   Do not open the plan.
3. Click **Actions** > **Duplicate**.
4. Enter a plan name and optional description.

   The default name is `Copy of <original name>`.

Data Set Permissions

Assign permissions to a data set to control the level of access to the data set.

When you run the plan that creates the data set, you become the owner of the data set. A data set owner and the domain administrator can add users and user groups and assign the required level of access. You can access data sets based on the permissions that you have. If you do not have the required permissions, you cannot open data sets that appear in search results.

Data sets have the following levels of permission:

- Read
- Write
- Execute

To perform a task that uses a data set, you must also have the minimum required level of privileges as a TDM user.

The following table lists the permission levels, the tasks that you can perform with each level, and the minimum required privileges for each task:

Permission	Description	Minimum Required Privilege
Read	- Open and view the data set. - Monitor logs for the data set. - View the data in tables in the data set - Create and delete data coverage tasks - Generate and execute a data set subset plan	- View Data Set - View Data in Data Set
Write	- Open and view the data set. - Monitor logs for the data set. - View the data in tables in the data set - Create a data set. - Edit the description and tags of the data set. - Classify tables in a data set. - Edit, delete, and add data in the tables in a data set. - Add and edit tags to data in a data set. - Delete a data set. - Perform Fill Cell jobs in data coverage tasks - Create a data set subset plan.	- View Data Set - View Data in Data Set - Manage Data Set - Manage Data in Data Set
Execute	- Open and view the data set. - Monitor logs for the data set. - View the data in tables in the data set - Reset a data set.	- View Data Set - View Data in Data Set - Reset Data Set

The data set owner and the administrator can lock and unlock a data set. You cannot edit or reset a locked data set.

Editing Data Set Permission

When you create a data set, you can provide read, write, and execute permissions to other users and user groups. Edit the permissions assigned to users and user groups from the **Permissions** tab of the data set.

1. Open the required data set and click **Permissions**.

   A list of the users and user groups with permissions for the data set appears.
2. Click **Edit** on the **Users** or **User Groups** tab.

   The **Edit Data Set Permissions** dialog box opens.
3. To edit the permission of a user or user group, select the user or user group from the list and edit the permission as required. You must save the changes for each user or user group.
4. To delete a user or user group, select the user or user group from the list and click **Delete**.
5. To add a user or a user group:
   a. Click **Add Users** or **Add User Groups**.
   b. Select one or more users or user groups.
   c. Optional. From the list of permissions, select the required permissions if either of the following statements is true:
      - You selected a single user or user group.
      - You want to assign the same levels of permission to all selected users or user groups.
   d. Click **OK**. TDM adds the users or user groups to the list.

e. Select each user or user group and assign the required permission levels. You must save the changes for each user or user group. Skip this step if you performed step c.

6. Click **OK**.

Locking and Unlocking a Data Set

To prevent users from editing or resetting a data set, you can lock the data set. You cannot make changes to the data or the metadata in a locked data set. The owner and the administrator can reset and edit a locked data set. Lock a data set if you need to check the data before you allow other users to edit or reset it. You can unlock the data set when you want to allow edits to the data set.

1. Open the data set and click the **Overview** or **Permissions** tab.
2. To lock the data set, click **Actions** > **Lock**.

   The data set appears as locked in the **Data Sets** view. The **Edit** options do not appear in a locked data set.
3. To unlock the data set, click **Actions** > **Unlock**.

   The status of the data set is updated in the **Data Sets** view. The **Edit** options appear in the data set.

Note: If you lock a data set that is published to the self-service portal, you restrict the tasks that you can perform on the data pack in the self-service portal.

The following users can perform tasks on a data pack if the data set is locked:

- Administrator users and data pack owners can perform tasks on the data pack.
- Shared users can edit the general properties of the data pack including the tags and description.

Monitor a Data Set Job

You can monitor all data set jobs in the **Monitor** view from the Test Data Manager home page. You can monitor jobs for a specific data set from the **Monitor** tab in the data set page.

You can view jobs created by any Test Data Manager Service in the domain. Identify the node based on the Test Data Manager Service listed in the properties.

You can monitor the following types of jobs for a data set:

- Created. Creates a data set in the test data warehouse.
- Reset. Resets a data set to a target.
- Generate Workflow. Generates a workflow from a data set.
- Execute Workflow. Runs a workflow to create a related data set after you generate the workflow.

You can manually refresh the jobs. You can remove a job that you scheduled to run at a later time.

View and Manage Data in a Data Set

You can view the data in a data set from the data set page. Based on user privileges and the level of permissions that you have on a data set you can edit the data in the data set.

Edit the data set to add or remove rows from a table or to update the data. Use the **SQL Query** field to edit data or directly edit the data in the grid. You can perform data manipulation language (DML) operations to edit the data. You cannot perform data definition language (DDL) operations.

You can add row-level tags to the data to track the changes. You can filter and search for data in a data set based on row-level tags that you add.

You can reset a data set with the edited data to a test environment to return the data to a required state.

Rules and Guidelines for Viewing and Managing Data

You can view and edit data in a data set from the **Tables** tab in a data set page.

Consider the following rules and guidelines when you view or edit data in a data set:

- The default SQL query is SELECT * FROM.
- You can use the WHERE clause with SELECT queries.
- You cannot run a SELECT query with aggregate functions such as COUNT, MIN, MAX.
- You cannot run a SELECT query with a GROUP BY clause.
- You cannot run a SELECT query with Joins.
- If a table or column name contains spaces or special characters, you must enter the name in double quotes (") when you use it in a query.
- To run a query that uses the WHERE clause with the date data type, you must enter TO_DATE in the SELECT statement.
  For example:

  ```
  SELECT * from INVOICES where INVOICE_DATE=TO_DATE('01/06/2003 00:00:00','mm/dd/yyyy hh24:mi:ss').
  ```
- You cannot run an SQL query with a table alias.
- You cannot run an SQL query that contains the table name with the owner name.
- You cannot insert numeric data with data values outside the range of -9223372036854775807 to 9223372036854775807.
- You cannot use a comma (",") in a tag as the comma is a tag delimiter. Use commas to separate tags.
- You cannot view or edit data in tables that contain the special character "." in the table name or column name.
- You cannot view and edit data from an Oracle database that is of BLOB and RAW data type.
- You cannot view or edit data from a data type of a database that is mapped to Oracle BLOB and RAW data types.

Viewing Data in a Data Set

You can view data in tables in a data set from the data set page.

1. Open the required data set.
2. Click the **Tables** tab.

3. Select the table for which you want to view the data.

   The table data appears in a grid on the **View Data** tab.

4. Optional. To view specific data, run an SQL query to select data based on required criteria.

   Data that matches the criteria appears in the grid.

5. Optional. If the data contains row-level tags, you can enter the tag name in the **Filter Tags** field and search for and view data based on tags. You can also select the required option from the **View By** list to view all tagged or untagged data.

   If you first run an SQL query to select data, the filters search for and return matching data from the results of the SQL query.

Using the SQL Query Field to View and Manage Data

You can run an SQL query to edit the data in a data set.

1. Open the required data set.
2. Click the **Tables** tab.
3. Select the table that you want to update.

   The table data appears in a grid in the **View Data** tab in the panel below the list of tables.

4. Optional. To view specific data, run an SQL query to select data based on required criteria.

   Data that matches the criteria appears in the grid.

5. Enter the SQL query to insert, delete, or update the data in the **SQL Query** field and click **Run**.

   You can view the updated data in the grid.

SQL Queries to View and Edit Data

You can run SQL queries to view and edit data in a data set.

The following table lists the queries that you can run along with sample syntax:

Query	Sample Syntax
SELECT *	`SELECT * FROM <table_name>;`
SELECT	`SELECT <column_name>,<column_name> FROM <table_name>;`
SELECT	`SELECT <column_name> AS <alias_name> FROM <table_name>;`
INSERT INTO	`INSERT INTO <table_name> (column1,column2,column3) VALUES (value1,value2,value3);` You must enter the column names. You cannot run the query without the column names.
UPDATE	`UPDATE <table_name> SET column1=value1,column2=value2`
DELETE FROM	`DELETE FROM table_name WHERE <column_name>=<value>;`
DELETE FROM	`DELETE FROM table_name;` Deletes all data in the table.

Updating Data in the Grid

You can update the data in required cells directly in the grid.

1. Open the required data set.
2. Click the **Tables** tab.
3. Select the table for which you want to update the data.

   The table data appears in a grid in the **View Data** tab in the panel below the list of tables.
4. Optional. To view specific data, run an SQL query to select data based on required criteria.

   Data that matches the criteria appears in the grid.
5. Click to select the row that you want to edit.

   The fields in the row become editable.
6. Click the required field and edit the data as required.
7. Click the **Save** button to save the changes.

   The updated data appears in the grid.

Adding Rows to a Data Set Table

You can add rows to a table in a data set.

1. Open the required data set.
2. Click the **Tables** tab.
3. Select the table that you want to update.

   The table data appears in a grid in the **View Data** tab in the panel below the list of tables.
4. Click the **Add Row** button in the right-hand corner of the panel.

   A new row appears in the grid.
5. Enter the data in the required data fields.

   You cannot add a blank row. Enter data in at least one column.
6. Click the **Save** button to save the row to the table.

   The data is added to the table and appears in the grid.

Deleting Rows in a Data Set Table

You can delete rows from a table in a data set.

1. Open the required data set.
2. Click the **Tables** tab.
3. Select the table that you want to update.

   The table data appears in a grid in the **View Data** tab in the panel below the list of tables.
4. Click the check box to select the rows that you want to delete.
5. Click the **Delete Row** button in the right-hand corner of the panel.
6. Click **OK** to confirm the delete.

   The rows are deleted from the table.

Managing Row-level Tags in a Data Set Table

You can add, delete, and update row-level tags to data in a data set.

To delete a tag that you enter in the **Filter Tags** field or the **Update Tags** dialog box, click the **X** icon that appears with the tag. To use the keyboard to delete a tag, press **Tab** or **Shift+Tab** to select the tag and then press **Delete** from the keyboard.

1. Open the required data set.
2. Click the **Tables** tab.
3. Select the table for which you want to update the data.

   The table data appears in a grid in the **View Data** tab in the panel below the list of tables.
4. Optional. To view specific data, run an SQL query to select data based on required criteria.

   Data that matches the criteria appears in the grid.
5. Optional. Use the **Filter Tags** field to view data with specific tags, or use the **Filter By** option to view all tagged or untagged data.

   Data that matches the criteria appears in the grid.
6. To select the rows in which you want to add, delete, or update tags, click the required check boxes.
7. Click the **Update Tags** button in the right-hand corner of the panel.

   The **Update Tags** window opens.
8. Add, delete, or update tags and then click **OK**.

   Use a comma to separate tags. The tags are added, deleted, or updated in the selected rows in the table.

CHAPTER 12

Analyzing Test Data with Data Coverage

This chapter includes the following topics:

- Data Coverage Analysis Overview, 250
- Data Coverage Process, 251
- Creating a Data Coverage Task, 251
- Data Coverage Task Columns, 252
- Data Coverage Analysis Page, 257
- Editing a Data Coverage Task, 258
- Marking a Cell as Invalid, 258
- Updating Data Across Cells, 259
- User Input in Fill Cell Jobs, 260
- Data Coverage Analysis Example, 262

Data Coverage Analysis Overview

You can analyze the data in a data set to check whether you have a minimum amount of data for specific test cases.

Create a data coverage task to perform pairwise data analysis and to create a visual representation of the data coverage in a data set. You can assess the quality of test data by analyzing combinations of values in any two columns. You can change the combinations to ensure that you cover all valid combinations of values. You can improve the quality of the data and move data across categories to meet the minimum data threshold that you require.

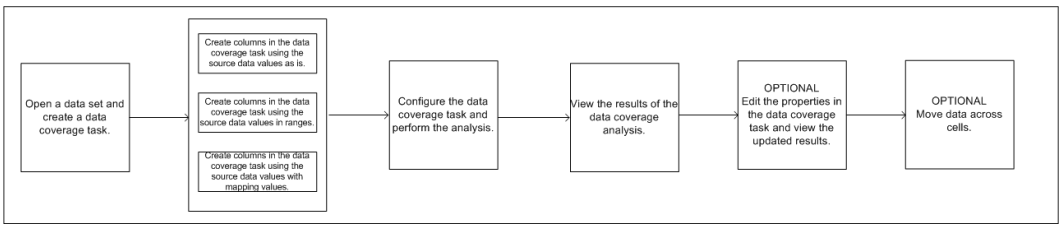
For example, you need to test a banking application that offers credit cards to customers. You create a data set with tables that contain data related to the credit card types and the criteria for each. The data could include location and the minimum balance required for each type of card. The data set also contains tables with customer information. To understand whether you have sufficient data for the different test cases, you need to analyze the amount of data that you have in different categories. For example, you need to know if you have sufficient data for each type of card in each location.

When you analyze the data, you also see if there is more data than what you require for some locations. You can then update the data records across columns or data ranges to ensure that you have sufficient data density for test cases.

Data Coverage Process

Create a data coverage task to analyze the data in a data set. You can edit the parameters used in the analysis. Based on the results, you can choose to move data values across ranges or groups.

The following image shows the tasks that you perform to create a data coverage task and the tasks that you can perform in the task:



You can perform the following tasks in a data coverage task that you run:

Create columns to use in the data coverage analysis

Create columns that map to source data columns to use in the analysis. You can create columns that use source data as is or in ranges of data values. You can also create columns that map data values across specific groups of values.

Run the data coverage task and view the results of the analysis on the data coverage page

You can run the data coverage task and view the results in the data coverage page. You can edit the results view to select and view specific cells in the results.

Edit the parameters used in the analysis

You can edit the parameters including the columns and filters applied and view updated analysis results in the data coverage page.

Update data values in the source data

You can update data values across cells that you analyze. For example, based on the data coverage analysis results, a cell contains data values below the minimum threshold that you set. You can update data in other cells to create the minimum required data values in the cell.

Creating a Data Coverage Task

Create a data coverage task to analyze the data coverage across combinations of data values, ranges, or groups.

1. Open the data set that contains the data that you want to analyze.
2. Click the **Data Coverage** tab.
3. Click **Actions** > **New**.

   The **New Data Coverage Task** window appears.
4. Enter a name and optional description for the data coverage task.
5. Enter the minimum threshold value.

   The minimum threshold represents the minimum number of values you require to consider the data density adequate.

6. Click **Next**.
7. Select the master table for the data coverage task and click **OK**.

   The related tables appear in the **Related Tables** list. You can remove tables that you do not want to include in the analysis.
8. Optional. Click the **Add Tables** button of a related table to add tables related to the table.

   You can enter an alias name if you want to use a different name to identify the table in the task. Alias names must be unique within a task. If you use a single table multiple times, use a different alias each time.
9. Click **Finish** to create the task.

   The data coverage page opens.
10. Configure the data coverage task. Create the columns that you want to use in the analysis.
11. From the **Columns** pane, select a column that you want to add to the X axis on the graph and click **Move to X**. You can also click the required column name from the list of columns and drag the column into the **X Axis** field.

    The column values are plotted along the X axis.
12. Select and add a column to the Y axis.

    The column values are plotted along the Y axis.
13. Optional. Edit the data count threshold.
14. Optional. Add a filter column to filter results based on column criteria.

    You can add two filters.
15. Click **Analyze**.

    The data coverage analysis results appear on the graph. Data coverage is represented based on the minimum and maximum data count values that you set. The color of the cell indicates the density of data. White indicates no data, light blue indicates data below the minimum threshold, and dark blue indicates a higher data density than the minimum threshold that you set.
16. Optional. Click the **X Axis** field or the **Y Axis** field and select specific cells from the list to view specific results.

Related Topics:

- "Creating a Data Coverage Column Using Data As Is" on page 255
- "Creating a Data Coverage Column Using Data Ranges" on page 255
- "Creating a Data Coverage Column Using Mappings" on page 256

Data Coverage Task Columns

Create a column in a data coverage task to analyze combinations of data values in a data set for density of data coverage. You can also create columns to use as filters in the data coverage analysis.

You can analyze and plot the data coverage in different ways based on the kind of data in the data set. You can use individual column values in the analysis or assign the data to ranges that you create. You can create mapping values and map the data values to mapping values.

You cannot use source columns with binary data type in a data coverage task.

You can create columns to use the data in the following ways:

Use as is

Use individual data values in the analysis to plot the data coverage of distinct values. Individual values are plotted in the graph in the data coverage task. Use data as is when you have a small number of distinct values in the column and you want to view data coverage for individual values. For example, low cardinality columns.

Range

Create ranges of values and analyze the data based on these ranges. For example, a table on employee information includes a Salary column. You want to analyze the data coverage for different salary values across different locations. You can create ranges for the salary values. The data coverage analysis indicates the data density for different salary ranges across locations.

You can use ranges for numeric and date data types

Mapping

Create mapping values to analyze data in groups. Map each of the data values to a mapping value. You can then use the mapping value in the analysis to plot data density across groups of values. For example, a test case requires data in a few regions. You therefore want to analyze the distribution of data across regions. The data contains a States column. You can create mapping values such as East, West, North, South, and assign states to a mapping value. You can then analyze the data distribution across regions.

You can map data values to a single mapping value. You can map multiple data values to the same mapping value.

Data Type Exceptions

You cannot create data coverage columns with source data columns that contain certain data types.

The following table lists the data types that you cannot use to create data coverage columns:

Data Coverage Column Type	Data Type
Use As Is	You cannot use the following data types to create data coverage columns that use data as is: - Date - Real - Double - Precision - Decimal - Decimal (p,s) - Float - Binary_float - Binary_double
Range	String
Mapping	You cannot use the following data types to create data coverage columns that use mappings: - Date - Real - Double - Precision - Decimal - Decimal (p,s) - Float - Binary_float - Binary_double

You cannot include the following data types in data coverage tasks:

- Dburitype
- Xdburitype
- Httpuritype
- Timestamp with local time zone
- Timestamp with time zone
- Urowid
- Day to second
- Year to month
- Nclob
- Longvarchar

Creating a Data Coverage Column Using Data As Is

Create a column using data values as is to analyze the data coverage for individual data values.

Review the list of data type exceptions before you create a data coverage column.

1. Open the data coverage page.
2. To add a column that you want to plot on the graph, click the **Add** button on the **Columns** pane.

   The **Add Column** window opens.
3. From the list of tables, select the table that contains the column.
4. From the list of columns, select the source data column to which this column refers.

   The column data appears in the **Data Preview** pane.
5. Enter an alias name for the column.

   Column aliases must be unique within a task.
6. Select the **Use as is** type.

   The data from the preview tab is added to the **Value** tab.
7. Optional. You can edit the data in the **Value** tab. Click the **Reset** button to replace the edited data with data from the preview pane. Click the **Add** or **Remove** button to add or delete individual values.
8. Click **OK**.

Related Topics:

- "Data Type Exceptions" on page 254

Creating a Data Coverage Column Using Data Ranges

Create a column using ranges to analyze the data coverage of data values across specific ranges. You can use ranges to analyze data that is in numeric or date data type.

Review the list of data type exceptions before you create a data coverage column.

1. Open the data coverage page.
2. To add a column that you want to plot on the graph, click the **Add** button on the **Columns** pane.

   The **Add Column** window opens.
3. From the list of tables, select the table that contains the column.
4. From the list of columns, select the source data column to which this column refers.

   The column data appears in the **Data Preview** pane.
5. Enter an alias name for the column.

   Column aliases must be unique within a task.
6. Select the **Range** type.
7. To create a range, enter the start value of the range in the **Start** field and the end value of the range in the **End** field.

   The start value must be less than the end value. There must be no overlap of values in different ranges.
8. Enter a label for the range in the **Label** field.
9. Click the **Add** button to add another range.
10. Repeat steps 7 to 9 to create the required number of ranges.

Related Topics:

- "Data Type Exceptions" on page 254

Creating a Data Coverage Column Using Mappings

Create a column using mappings to analyze the data coverage of data values across specific groups of data values.

Review the list of data type exceptions before you create a data coverage column.

1. Open the data coverage page.
2. To add a column that you want to plot on the graph, click the **Add** button on the **Columns** pane.

   The **Add Column** window opens.
3. From the list of tables, select the table that contains the column.
4. From the list of columns, select the source data column to which this column refers.

   The column data appears in the **Data Preview** pane.
5. Enter an alias name for the column.

   Column aliases must be unique within a task.
6. Select the **Mapping** type.

   The data from the preview tab is added to the **Value** tab on the left.
7. Optional. You can edit the data in the **Value** tab on the left. Click the **Edit** button to edit the values. Click the **Reset** button to replace the edited data with data from the preview pane.
8. To create a mapping value, click the **Edit** button in the mapping values panel on the right.
9. In the **Value** field, enter a mapping value name and click the **Add** button to create another value field.
10. Repeat step 9 to create the required number of mapping values.
11. Click **OK**.
12. In the **Add Column** window, select a data value in the data panel on the left.
13. From the mapping values panel on the right, select the mapping value to which you want to map the data.
14. To link the data value to the mapping value, click the **Link** button.

    An arrow appears to indicate the mapping value that the data value is linked to. You can use the **Unlink** button to remove a link.
15. Repeat steps 12 to 14 to link each data value to a mapping value.
16. Click **OK**.

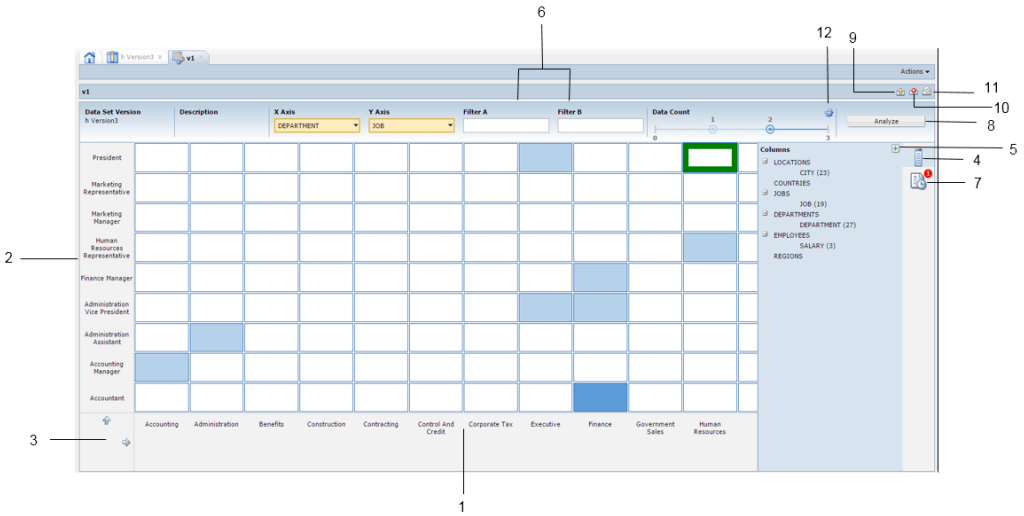
Related Topics:

- "Data Type Exceptions" on page 254

Data Coverage Analysis Page

A data coverage analysis page plots the data in pairs of columns in a graph. The results display the data coverage for combinations of values.

The following image shows a data coverage analysis page:



The following list describes the properties and options on a data analysis page:

1. X axis. The X axis plots one of the columns that you use in the analysis. In the image, values in the DEPARTMENT column are plotted along the X axis.
2. Y axis. The Y axis plots the second column that you use in the analysis. In the image, values in the JOB column are plotted along the Y axis.
3. Scroll buttons for the X and Y axes. The scroll button is visible when the values that you plot exceed the space available on the page.
4. Slider. Click the slider button to expand or collapse the list of data coverage task columns.
5. Add columns. Click the Add Columns button to create a data coverage task column.
6. Filter columns. Data coverage task columns that you can use as filters to further configure the analysis. You can add up to two filter columns.
7. Jobs pending. Click the Jobs Pending button to display a list of jobs that require user input to complete.
8. Analyze. Click the Analyze button to perform the analysis based on columns that you add to the X and Y axes.
9. Fill cell. Select a cell on the graph and click the Fill Cell button to update the data count in the cell. You can then select the cells from which you want to edit the data count.
10. Mark as invalid. Select a cell and click the Mark as Invalid button to exclude a cell from the analysis. The cell becomes unavailable in the graph. Use this option for cells that represent combinations of data that might not be valid for the analysis.

11. Mark as valid. Select a cell that you marked as invalid and click the Mark as Valid button to include the cell in the analysis.
12. Data count settings. Click the Data Count settings button to edit the minimum and maximum data count values for the analysis. You can also drag the markers along the data count bar.

Editing a Data Coverage Task

Edit a data coverage task to update the metadata or edit the data included in the task. You can edit a data coverage task to continue analysis or update the data coverage analysis. If you edit a data coverage task, the previous analysis information is replaced with the updates.

1. Open the data set that contains the data coverage task that you want to edit.
2. Click the **Data Coverage** tab.
3. Click the data coverage task that you want to edit.

   The data coverage page opens. The graph displays the results of the last analysis that you performed.
4. Optional. Click **Actions** > **Edit** to edit the name or description. Click **OK**.
5. Optional. Click **Actions** > **Edit** > **Tables** to edit the master table or the related tables in the task. Click **OK**.
6. Optional. Edit the column information as required. You can create and delete columns.
7. Edit the data coverage task. From the **Columns** pane, select a column that you want to add to the X axis on the graph and click **Move to X**. You can also click the required column name from the list of columns and drag the column into the **X Axis** field.

   The column values are plotted along the X axis.
8. Select and add a column to the Y axis.

   The column values are plotted along the Y axis.
9. Optional. Edit the data count threshold.
10. Optional. Add or edit a filter column to filter results based on column criteria.

    You can add two filters.
11. Click **Analyze**.

    The data coverage analysis results appear on the graph. Data coverage is represented based on the minimum and maximum data count values that you set.

Marking a Cell as Invalid

Some cells on a data analysis visualization might not be applicable for the analysis. You might not require the data that the cell represents. You can mark a cell that you do not want to consider in the analysis as invalid. You can change this if you want to include the cell in the analysis.

1. Open the data set that contains the data coverage task that you want to edit.
2. Click the **Data Coverage** tab.
3. Click the data coverage task that you want to edit.

   The data coverage page opens. The graph displays the results of the last analysis that you performed.

4. Select the cell on the graph that you want to mark as invalid. The cell is highlighted with a green border.
5. Right-click the selected cell and click **Mark as Invalid**.

   The cell greys out to indicate that it is invalid. This does not affect the results in other cells.
6. Optional. To mark the cell as valid, right-click the selected cell and click **Mark as Valid**.

Updating Data Across Cells

Based on the requirement and data coverage, you can update data across cells to add data where you need it or remove excess data. The data is updated in the data set when you edit data across cells in the data coverage task.

Note: Based on the schema, a move operation might fail in some cases. You cannot update data in a cell that contains data of the Blob data type.

1. Open the data set that contains the data coverage task that you want to edit.
2. Click the **Data Coverage** tab.
3. Click the data coverage task that you want to edit.

   The data coverage page opens. The graph displays the results of the last analysis that you performed.
4. Select the cell on the graph to which you want to add data. The cell is highlighted with a green border.
5. Right-click the selected cell and click **Fill**.

   The **Fill Cell** dialog box opens.
6. Select the source cell from which you want to move data on the graph. You can select multiple source cells.

   The **Fill Cell** dialog box lists the source cells that you select and suggests the number of rows to move. You can enter a different number.
7. Enter the number of rows that you want to move from each source cell.
8. Click **Fill**.

   A separate job runs to fill data from each of the source cells.
9. If the data contains fields that need input, a dialog box opens with fields to enter the data. Enter the data and click **OK**.

   The data from the source cells is edited to fill the selected cell. If you close the task before the user input dialog box appears, the job pauses. When you open the task, you must click the **User Input** button to view and enter the required data.

You can verify that the data is updated correctly after the task completes. Click the **Analyze** button without making any changes to the task. The results plotted on the graph must not change.

User Input in Fill Cell Jobs

You can update data across cells in a data coverage task to create data where needed. You might need to enter data in some situations where data is missing.

You can compare combinations of data from a single table or combinations of data from different tables. The X and Y axis can therefore represent columns from the same table or from different tables.

When you compare columns from the same table, a fill operation from one cell to another cell moves data that exists in a single table. The data is available to move and the fill operation runs without the need for input.

When you compare columns from different tables, some records in intermediate tables might not exist in the database. These records might be required for the combination corresponding to the target cell.

When there is no data available to fill a cell, you must enter the data values manually. The job pauses when input is required. If the data coverage task is open, a dialog box appears with fields to enter data values.

Enter data values that match the column data type. A move operation fails if the data that you enter does not match the column data type.

User Input Example

You want to analyze the data for employees in an organization. You want to analyze data for the different paygrades across regions. You create a data set that contains information on employee paygrades and organization branch and regions.

Example Data

Consider the following data set tables:

Table 1. REGION_DATA

REGION_ID	REGION_NAME
1	East
2	West

REGION_ID	REGION_NAME
3	North
4	South

Table 2. BRANCH_DATA

BRANCH_ID	BRANCH_NAME	REGION_ID	IFSC	FACILITY
102	Jaipur	3	1545	Large
105	Bangalore	4	6765	Small
103	Calcutta	1	1421	Medium

Table 3. EMP_INFO

EMP_ID	PAYGRADE	BRANCH_ID
1	Low	105
2	Medium	102
3	Medium	105
4	Medium	103
5	Low	105

Data Coverage Analysis

You create a data coverage task and plot the regions on the X axis and the paygrade column on the Y axis.

The data that you plot creates the following data on the graph:

Region/Paygrade	Low	Medium	High
East	0	1	0
West	0	0	0
North	0	1	0
South	2	1	0

There is more than one record in the cell that represents low paygrade in the south. The test cases that you run do not require data for low paygrades. You want to move data to the cells that represent high paygrade in the west and medium paygrade in the east.

You run a Fill Cell job to fill data in the cell that represents medium paygrade in the east. Choose to move one record from the cell that represents low paygrades in the south.

The Move operation updates the EMP_INFO table. One record that has a Branch ID that represents the south and a low paygrade is edited. The move updates the paygrade to Medium and the BRANCH_ID to 103. No user input is needed as all the other data required is available.

You run a Fill Cell job to fill data in the cell that represents high paygrade in the west. Choose to move one record from the cell that represents low paygrades in the south.

To perform the Move operation, the EMP_INFO table must be updated. The operation must add a record that contains a branch ID for the west. The branch ID in the EMP_INFO table is a foreign key that points to the BRANCH_DATA table. The BRANCH_DATA table does not contain a record for the west region. A record for the west region must be added to the BRANCH_DATA table. The operation adds a record to the BRANCH_DATA table and populates the region ID as 2 for West. The BRANCH_ID information and other information cannot be populated and must be manually entered.

A user input dialog box appears with fields to enter values for the following columns:

- BRANCH_ID
- BRANCH_NAME
- IFSC
- FACILITY

Therefore the Move operation updates two tables. A row is added to the BRANCH_INFO table. Data is modified in the EMP_INFO table.

Data Coverage Analysis Example

You work with a QA team that tests an ERP software product.

The team uses TDM to create data sets to store and manage the test data. Before you begin testing the next version of the product, you want to know if a data set contains data required to run specific test cases.

To run test cases, you need sufficient salary data for specific locations. You also need data in specific departments in each location. You need at least 50 records in each department across locations.

Tables in the Data Set

You create a data set ERP_CL1_2.0.

The data set contains the following tables:

EMP_DETAILS

The table contains the following columns:

- EMP_NAME
- EMP_ID
- EMP_DEP
- DEP_ID
- EMP_DOB
- EMP_ADDRESS

EMP_SAL_INFO

The table contains the following columns:

- EMP_NAME
- EMP_ID
- EMP_DEP
- EMP_SAL

ORG_INFO

The table contains the following columns:

- DEP_NAME
- DEP_LOC
- DEP_ID
- DEP_MGR

ORG_REV_INFO

The table contains the following columns:

- LOC
- DEP_NAME
- DEP_REV

Analysis for Data Coverage

Create a test data coverage task to analyze the data coverage in the data set.

Perform the following high-level steps to configure the data coverage task and view the results:

1. Create a column COL1 that refers to the EMP_SAL column in the EMP_SAL_INFO table. Use the Range type to plot the data.
2. Create a column COL2 that refers to the LOC column in the ORG_REV_INFO table. Use the Use as is type to plot the data.
3. Create a column FILTERCOL1 that refers to the DEP_NAME column in the ORG_REV_INFO table. Use the Use as is type to plot the data.
4. Plot COL1 along the X axis and COL2 along the Y axis.
5. Configure the minimum and maximum data values as 50 and 75.
6. Click Analyze to view the data coverage plotted across the graph. The results depict the data coverage for the salary ranges across locations.
7. To view the data coverage information for specific departments, add the column FILTERCOL1 to the Filter A field and click Analyze to refresh the results.
8. Select the required departments from the filter list to view results for individual departments.

Consider the following example:

The data coverage results indicate that there is no data for some departments for the salary range 5,000 - 10,000 for the location UK. However, there is data well above the maximum threshold for the salary range 15,000 - 20,000 in the location US. There is some data in the salary range 20,000 - 25,000 in the location AUS. This data is slightly above the minimum threshold value. You want to update some data from these cells to the 5,000 - 10,000 in UK cell.

Perform the following high-level steps to configure the data coverage task and view the results:

1. Select the cell that represents the data for the location UK and salary range 5,000 - 10,000.
2. Right-click and select Fill.
3. Select the cell that represents the US location and salary range 5,000 - 20,000.
4. Select the cell that represents the salary range 20,000 - 25,000 in the location AUS.
5. The Fill Cell dialog box adds these fields and suggests the number of rows that you might add from each cell.
6. You can edit the number in the field to move the required number of rows. You cannot move more rows than what exists in a cell.
7. Click Fill. If user input is required, a dialog box appears with fields to enter data values. For example, you might need to update parent records in some tables.

For example, when you update the data from the US and AUS locations to the UK location, the UK location might not contain any department information. In this situation you enter data values for the department rows.

You can configure the data coverage task to analyze the data based on your requirement. You can then update the data to create data in rows where you need the data.

CHAPTER 13

Plans and Workflows

This chapter includes the following topics:

- Plans and Workflows Overview, 265
- Workflow Connections, 267
- Plan Components, 267
- Pre Workflow and Post Workflow Parameters, 268
- Plan Settings, 268
- Masking Components, 280
- Subset Components, 280
- Generation Components, 281
- Component Criteria, 281
- Source Settings, 282
- Plan Management, 290
- Workflow Generation, 293
- Parameter Files in Test Data Manager, 294
- Executing a Workflow, 295
- Workflow Executions View, 296

Plans and Workflows Overview

A plan defines a data subset, data masking, or data generation operation. It includes the components that you need to generate a workflow.

When you create a plan, you add data masking, data subset components, or data generation components based on the operation that you need to perform. You cannot add data masking or data subset components to a plan when you add data generation components. You can add policies and rules to run data masking operations. Add groups to run data subset operations on relational or flat file sources.

Add XSD files if you want to perform data subset or data masking on XSD sources. Add entities and tables or XSD files to run data generation operations.

If you add both subset and masking components to a plan, the subset component gets priority. TDM performs the subset operation first. TDM then applies masking rules to columns in the subset data that have masking assignments. TDM does not apply masking assignments to source data that is not included in the subset, even if the data is part of a cascade.

Define the workflow connections in the plan. Define the source and target connections. If required for the plan, edit staging and storage connections and define lookup and dictionary connections.

When you update or delete a plan component, you must generate and run the plan again to update the changes.

You can use parameters in a plan to vary the output of a workflow without updating plan components. Choose the parameter file location or parameter values when you run the workflow.

To copy flat file results to a test tool that you have integrated with TDM, configure the test tool properties in the plan.

To store data in the test data warehouse, you select the test data warehouse as the target in the plan.

You can create multiple workflows from one plan. Define workflow properties in the plan, such as commit properties, update strategies, and recovery strategies.

View the sessions that run from each workflow. View statistics that describe the number of rows that the target receives. The **Session Details** tab shows the affected rows, the rejected rows, and the number of rows that the session wrote to the target.

When you start a workflow, the PowerCenter Integration Service completes the plan operations.

Plans and Workflows Task List

Perform the following steps to create the plan and run the workflow:

1. Create a plan. Enter a plan name and description.
2. Add data masking rules and policies to the plan.
3. Add data subset entities, groups, and templates.
4. Add data generation entities and tables to the plan. When you enter data generation components, you cannot enter data masking or data subset components.
5. Switch masking rules off or on.
6. Optionally, change the criteria of data subset components.
7. Configure plan settings including the plan properties, error and recovery strategies, and advanced settings.
8. To store data in the test data warehouse, select the test data warehouse as the target connection.
9. To copy flat file results to a test tool that you have integrated with TDM, configure the test tool properties in the plan.
10. Optionally, configure an override strategy for a data source or a table.
11. Optional. If the plan contains parameters, download a parameter file to review parameters used in the plan and the parameter values.
12. Generate a workflow. You can choose to generate and run a workflow in a single step.
13. If the plan contains parameters, choose the parameters and parameters to use when you run the workflow.
14. Run the workflow.
15. Monitor the workflow.

Workflow Connections

Workflows contain connections to the PowerCenter Integration Service, the TDM repository, and one or more connection objects.

An offline job requires three connections to the TDM repository. When you run a plan with multiple workflows, each workflow requires one connection. You can run a maximum of eight workflows at a time. When you run eight workflows, you need 11 connections. Additional workflows move into the queued state.

Each workflow can use different connections to relational sources and paths to flat files. Choose the connections in the plan.

When you create a plan, you can select the following connections:

- Source. The connection that is used to connect to the source.
- Target. The connection that is used to connect to the target.
- Staging connection. A connection to a database for staging data. The default connection entered in the Preferences view appears. You can change the connection for a plan.
- Lookup connection. A connection to a database that contains lookup tables.
- Storage connection. A connection to a database that contains storage tables. The default connection entered in the Preferences view appears. You can change the connection for a plan.
- Teradata connection. An ODBC connection to a Teradata database.
- Dictionary connection. A connection to a database that contains a dictionary table for substitution, email masking, and data generation.

Plan Components

When you create a plan, you add components to the plan based on the operations that you want to perform.

You can perform data masking operations, data subset operations, or both. You can also perform data generation operations. When you perform data generation operations, you cannot perform data subset or data masking operations.

The following table describes the components that you can add to a plan for each type of operation:

Component	Description
Rule	Data masking rules. For a data masking operation, you can add rules assigned to a policy and add rules that are not assigned to a policy. You can add rules to a data masking plan.
Policy	A Policy is a set of data domains. Each data domain can have multiple rules. You can add policies to a data masking plan.

Component	Description
Entity	Defines a set of tables that are related based on physical or logical constraints. An entity can contain parent tables and child tables. An entity maintains relational hierarchy in the tables. Define filter parameters for ports in the entity to extract the columns from the tables in an entity. When you run a workflow generated from a data subset plan, the PowerCenter Integration Service extracts source data based on the filter parameters defined in the entity and loads the data to the subset tables. If an entity is part of a template that is in the plan, you cannot add the same entity to the plan. Remove the template from the plan, or remove the entity from the template to continue. You can add entities to a data subset or a data generation plan.
Group	Defines a set of tables to copy to a target subset database. You can add groups to a data subset plan.
Table	Defines the target table columns that contain generation rule assignments. You can add tables to a data generation plan.
XSD File	The XML source definition that you import from the PowerCenter repository. Add to a plan if you want to perform data subset or data masking on an XML source or if you want to generate XML data. You can add XSD files to a data subset, data masking, or data generation plan.

Pre Workflow and Post Workflow Parameters

In a TDM plan, you can select workflows that you want to run before and after the current workflow.

Configure the pre workflow parameter if you want to run a specific workflow before you run the workflow from the current plan. Configure the post workflow parameter if you want to run a specific workflow after you run the workflow from the current plan. You can select an existing workflow in the PowerCenter folder that is associated with the project.

The pre workflow step is the first step in the process. The actual TDM workflow starts after the pre workflow completes successfully. The post workflow starts after all the TDM workflows complete successfully.

For example, you want to move files from the production area to the test area before you mask them. After you mask the data, you want to run statistics to optimize the tables. First, the pre workflow copies files from one location to another. Then TDM workflow masks the data. After TDM loads the masked data into the target, the post workflow runs the statistics.

You can disable triggers in a pre workflow and then enable them in a post workflow.

Plan Settings

Configure plan settings that apply to all data sources in the plan.

The plan settings include PowerCenter options, connection information, test data warehouse settings, test tool integration information, recovery strategies, and source and target properties.

Connection Options

Enter source and target connections.

The following table describes connection options:

Connection Options	Description
Treat the subset criteria as a variable	Appears if you select at least one subset component in a plan. Use the subset criteria as a variable only when there is an entity with a parameter value instead of a specific value in the entity criteria. When you select this option, you must enter the parameter file name and parameter file path.
Use Source Connection as Target Connection	Use the same connection for source and targets. Use with inplace masking.
Use Sorter Transformation	Use a Sorter transformation in the mapping to sort data. Required if the source and staging databases use different sorting techniques. You cannot use the sorting technique if you select a Hadoop HDFS connection.
Use Batch Updates for Recursive Executor	Update rows in batches in a data subset operation on tables with cyclic or tending to cyclic constraints.
Source Connection	A connection to the source database. Choose a variable, relational, or Hadoop HDFS connection from the list. Select the variable name or select a source connection from the list. You cannot use a Hadoop HDFS connection as a variable when you configure a source.
Source Connection Directory Path	Displays if flat file sources are present. Enter the path to the flat file source directory.
Target Connection	Choose a variable, relational, test data warehouse, flat file, or Hadoop HDFS connection from the list. Select a target connection or a variable name from the list. You cannot use a Hadoop HDFS connection as a variable when you configure a target. If you enter the target connection as a variable, ensure that you define the scope of the variable as **Global** or **Integration Service** in the parameter file. In a multinode setup, if you use parameters to enter the connection information, the parameter file must be present on the node on which the Test Data Manager Service runs. The file must be in the same directory path on both nodes.
Target JDBC Connection	Appears if you select an ODBC target connection type. The connection that contains the JDBC connection string of the ODBC target database. Required if the target connection type is ODBC and either of the following statements is true: - You want to disable and enable constraints in the plan. - The source connection type is ODBC and you want to generate mappings with the source data type. Optional if the target is ODBC and you want to perform simple data movement operations.
Source File Directory	Appears if the project includes XML sources. Location of the source XML file. Enter the path to the XML source file directory. The source file must reside on the machine on which the PowerCenter Integration Service runs. Default is the PowerCenter Integration Service process variable directory $PMSourceFileDir.
Target File Directory	Appears if the project includes XML sources. Enter the path to the XML target file directory. The path must be on the machine on which the PowerCenter Integration Service runs. Default is the PowerCenter Integration Service process variable directory $PMTargetFileDir.

Connection Options	Description
Source File Name	The name of the IMS source file or VSAM source file. If the source is a Hierarchical IMS file, enter the name of the source unload file. If the source is a Hierarchical VSAM/SEQ file, enter the name of the source file. Required if the source connection is PXW_NRDB Batch.
Target File Name	The name of the IMS target file or VSAM target file. If the target is a Hierarchical IMS file, enter the name of the target unload file. If the target is a Hierarchical VSAM/SEQ file, enter the name of the target file. The target file name must be different from the source file name. Required if the target connection is PXW_NRDB Batch.
Teradata ODBC Connection	Name of the Teradata ODBC connection. Required if the target connection is Teradata PT. Select a connection from the list.
Dictionary Lookup Connection	Connection to a database that contains lookup tables. Required if the plan has a dictionary generation rule assignment or a mapplet rule that includes a transformation. Choose relational or flat file from the list. Select a dictionary lookup connection from the list.
Dictionary Connection	A connection to the database that contains the dictionary table. The dictionary connection option does not appear unless the project contains a rule that requires a relational dictionary.
Email Dictionary Connection	A connection to the database that contains the email dictionary table. The email dictionary connection option does not appear unless the project contains a masking rule that requires a relational email dictionary.
Number of Records to Generate	Required. The number of records that you want to generate in the target table when you create a generation plan.
Reference Lookup Connection	A connection to the database that contains parent tables. Required if the generation plan has reference lookup rule assignment. Choose relational or flat file from the list. Select a dictionary lookup connection from the list.

Dictionary Connections

A dictionary is a flat file or relational table that contains substitute data. When you define a relational dictionary, you can define the connection to the dictionary in the project plan and workflow.

The **Dictionary Connection** field appears in the **Plan Settings** if the plan requires a connection.

When you define a substitution masking rule with a flat file dictionary, you can browse to a dictionary from Test Data Manager. Test Data Manager imports the structure of the dictionary file to enable you to define the serial number column and the lookup column.

By default, when you run a workflow, the PowerCenter Integration Service finds flat file dictionaries in the following location: `<Informatica installation directory>\server\infa_shared\lkpfiles`

You cannot change this location. The TDM installation installs the built-in dictionaries in the lkpfiles location. When you create a custom dictionary file, Test Data Manager copies the file to the `lkpfiles` directory.

Note: To access a dictionary on Microsoft SQL Server, create an ODBC connection using the DataDirect SQL Server Wire Protocol driver.

Data Set Properties

Configure the data set properties to use the test data warehouse as the target connection in a plan.

The data set properties appear when you select the test data warehouse from the list of target connections.

The following table describes the test data warehouse settings:

Property	Description
Name	Required. Name of the data set.
Description	Enter a description for the data set.
Data Set Version	The data set version. You cannot edit the version.
Application	The application name.
Application Version	The application version.
Tags	Tags for the data set. Enter the required tags separated by commas.

Test Tool Integration Options

Configure test tool integration in a plan to copy the flat file target results to a specific location in an HP ALM server. The administrator must first enable test tool integration for the TDM setup. You must run the workflow from Test Data Manager. The test tool integration job does not run when you run the workflow from the PowerCenter client.

The following table describes the test tool integration properties that you can configure:

Test Tool Integration Options	Description
Attach to Test Tool	Select this option to copy the flat file target results to a location in the HP ALM server integrated with TDM.
Project	Required. Browse to select the project in the HP ALM server where you want to store the flat file results.

File Path Settings

Configure the parameter file and file path settings in the **File Path Settings** section.

The file path settings appear if one or more of the following conditions are true:

- You select variable source or target connections.
- You choose to treat the subset criteria as a variable.
- An entity or group that you use in the plan contains a criteria parameter.
- You enter the seed value as a parameter or variable in a masking rule that you use in the plan.

The following table describes the file path settings:

Property	Description
Parameter File Name	The name of the parameter file that contains the parameter names and parameter values. Enter the name of the parameter file.
Parameter File Path	The path to the file that contains the parameter names and parameter values. Enter the path to the parameter file.

Target Options

Configure properties for committing data and error recovery in the **Target** fields.

The following table describes target options:

Other Property Options	Description
Truncate Table	Truncates the table before loading it. By default, this is not selected.
Disable Indexes	Disables indexes for faster loading.
Disable Constraints	Disables physical constraints in the database before loading the data to the target and enables the constraints after the data loads. If TDM fails to enable the constraints after loading the data, the workflow fails. You must then manually enable the constraints in the database and run the workflow again. **Note:** If a table in a PostgreSQL database contains an exclusion constraint and you select the **Disable Constraints** option, TDM does not disable the constraint even though the task completes successfully. **Note:** When the target connection is DB2 for zOS and if you choose to disable constraints, TDM drops and recreates the constraints and moves the data. But the database tablespace goes into Check Pending state and you cannot perform operations on any of the tablespaces. You must run the Check Data utility in the current database tablespace.
Disable Triggers	Disables DML triggers on the target database before loading the data to the target and enables the triggers after the data loads. If you enter the target connection type as relational, the **Disable Triggers** option gets selected when you choose the **Truncate Tables** option. If you enter the target connection as a parameter, you must manually select the **Disable Triggers** option. **Note:** This option disables only DML triggers.
Optimize Relationship Type for Entities	Optimizes relations in an entity at the plan level. Optimizes relations to ensure a minimal number of major relations such that the operation includes all tables in the entity. **Note:** You must have provided criteria at the plan level when you select this option.
Use Netezza Bulk Reader	Uses Netezza bulk reader to read data from a Netezza source connection. Appears if the plan contains a Netezza source connection.
Use Netezza Bulk Writer	Uses Netezza bulk writer to write data to a Netezza target connection. Appears if the plan contains a Netezza target connection.

If you select a Hadoop HDFS target connection, you cannot configure the target options.

Update Strategy Options

Configure how the PowerCenter Integration Service updates the target in the plan update strategy options. Configure whether the source rows are insert or update rows. Determine how the PowerCenter Integration Service updates the target based on whether the rows exist in the target.

The following table describes the options for update strategy:

Update Strategy Option	Description
Treat source rows as	Indicates how the PowerCenter Integration Service treats source rows. Choose Insert or Update. If an imported PowerCenter mapplet contains a transformation that is configured to set the update strategy, the default option is Insert. If you select the same connection as source and target, the option changes to Update. **Note:** If you use the same connection parameter for source and target, the option does not change to Update. You must manually set the value to Update or select the **Use Source Connection as Target Connection** option. If you select the **Use Source Connection as Target Connection** option, the value changes to Update.
Update as	Determines whether the PowerCenter Integration Service updates or inserts rows. Choose one of the following options: - Update. The PowerCenter Integration Service updates all rows flagged for update. - Insert. The PowerCenter Integration Service inserts all rows flagged for update. - Update else Insert. The PowerCenter Integration Service updates rows flagged for update if they exist in the target, and inserts remaining rows marked for insert.

Data Generation Properties

Before you create a plan to generate data, you can define default settings in the **Administrator | Preferences** view. You can update the default settings when you create a data generation plan.

The default rules are not populated if the rules are not part of a particular project. If the default rules are blank, then you need to select the default rules that are applicable to the plan.
The following table describes the properties to update default settings for data generation:

Property	Description
Default Numeric Generation Rule	The default rule you have set to generate data for numeric datatype. You can select a separate default rule for numeric datatype.
Default String Generation Rule	The default rule you have set to generate data for string datatype. You can select a separate default rule for string datatype.
Default Date Generation Rule	The default rule you have set to generate data for date datatype. You can select a separate default rule for date datatype.

Property	Description
Default Binary Generation Rule	The default rule you have set to generate data for binary datatype. You can select a separate default rule for date datatype.
Apply Default Rule to	The default rule you have set for all the columns without generation rules. You can also apply the default rule to the columns that do not have null values.
Minimum	The minimum number of child records for a parent record. Default is 1. You can enter a minimum numeric value.
Maximum	The maximum number of child record for a parent record. Default is 1. You can enter a maximum numeric value.
Parent Records Without Children	The number of parent records that have no child. Enter the percent value for self-cyclic tables or reference lookup generation rules.
Child Records Without Parent	The number of child records that have no parent. Enter the percent value for self-cyclic tables or reference lookup generation rules.

Error and Recovery Options

Configure properties for committing data and error recovery in the **Error and Recovery** fields. The section is minimized by default. The options are populated with default data if available.

The following table describes the error and recovery options:

Error and Recovery Options	Description
Stop on Error	Indicates how many non fatal errors the PowerCenter Integration Service encounters before it stops the session. If you enter zero, the session does not stop for non fatal errors. Default is zero.
Suspend on Error	Suspend workflow processing when an error occurs. You can fix errors and then restart the workflow. The default is No.
Rollback Transactions on Error	Rolls back the transaction at the next commit point when it encounters a non fatal write error. The default is Yes.
Recover Strategy	Strategy for recovering a workflow when errors occur. Choose one of the following recovery strategies: - Resume from last checkpoint. The PowerCenter Integration Service saves the session state of operation and maintains target recovery tables. - Restart task. The PowerCenter Integration Service runs the task again when it recovers the workflow. - Fail task and continue workflow. Default. The PowerCenter Integration Service cannot recover the task, but it continues the workflow.
Enable High Availability Recovery	Select Yes to enable workflow recovery. High availability recovery is not available for web service workflows. The default is No.
Commit Type	Choose a source-based or target-based commit. By default, the PowerCenter Integration Service performs a target-based commit.
Commit Interval	The number of rows to process before a commit. Default is 10,000.

Error and Recovery Options	Description
Error Log Type	Specifies the type of error log to create. You can specify flat file or none. Default is None.
Error Log File Directory	Required if you enable flat file logging. Specifies the directory where the log file is stored. Default is $PMBadFileDir\.

Encryption Options

Configure the encryption options to use the format preserving encryption type in a plan.

The encryption options appear when you configure format preserving encryption rule on the source data.

The following table describes the encryption options:

Encryption Option	Description
Pass Phrase	Select a pass phrase from the list of available pass phrases. The pass phrase value generates the key to encrypt or decrypt data in a format preserving encryption masking operation.
Encryption/ Decryption	Select encryption when you encrypt data. Select decryption when you decrypt data. **Note:** When you decrypt data, you must use the same pass phrase value and the same format preserving encryption rule configuration that you used to encrypt the source data.

Advanced Options

The Advanced Options include options to set the locale, change the staging connection and storage connection, and set the number of tables to process in a workflow. You can also configure an email address to send notifications to. This section is minimized by default. The options are populated with default data if available.

The following table describes advanced options:

Advanced Options	Description
Staging Connection	A connection to a database for creating temporary staging tables. The default set in the Workflow Generation section appears. You can change the connection used in a plan. Use this connection for scenarios that involve Cyclic Subset, Tending to Cyclic subset, Inplace Batch Updates, Non Relational Shuffle, and Hierarchical PWX. It is recommended that you use the source connection as the staging connection.
Storage Connection	A connection to a database that contains storage tables. The Data Masking transformation maintains the storage tables for repeatable substitution and expression masking between sessions. A storage table row contains the source column and a masked value pair. The storage tables for substitution and expression masking are separate tables, IDM_SUBSTITUTION_STORAGE and IDM_EXPRESSION_STORAGE.

Advanced Options	Description
XML date-time format	Appears if the source is an XML file. Date-time format defined in the session properties. Choose one of the following values: - Local Time. The time zone on the machine on which the PowerCenter Integration Service runs. - Local Time with Time Zone. The difference in hours between the PowerCenter Integration Service time zone and Greenwich Mean Time. - UTC. Greenwich Mean Time. Default is Local Time.
Datetime Format String	Date-time format defined in the session properties. You can enter seconds, milliseconds, microseconds, or nanoseconds. - Seconds. MM/DD/YYYY HH24:MI:SS - Milliseconds. MM/DD/YYYY HH24:MI:SS.MS - Microseconds. MM/DD/YYYY HH24:MI:SS.US - Nanoseconds. MM/DD/YYYY HH24:MI:SS.NS Default is microseconds.
Enable High Precision	Processes the Decimal datatype to a precision of 28. Default is No.
Tables Per Workflow	Sets the number of tables to process in a workflow. Default is 10.
Mapplet	Use to Reuse or Replace the mapplets in PowerCenter.
Max Parallel Sessions	The maximum number of sessions that can run at the same time. Applicable if you disable indexes and constraints in the target by selecting the options under Target options.
Pre-Workflow	The workflow that you want to run before you run the workflow from the current plan.
Post-Workflow	The workflow that you want to run after you run the current workflow from the current plan.
Locale	Sets the locale. A locale properties file corresponding to the locale you select must exist in the `<Informatica installation directory>/TDM/lang` location.
Data Integrity for Subset	Sets the integrity level for major constraints in the source data. Choose one of the following options: - Referential Integrity only. Ensures that the subset operation maintains referential integrity. This might fetch minimal data but might break transactional integrity. - Transactional Integrity for Major Relation Type. Ensures that the subset operation maintains referential and transactional integrity. This might fetch additional data, but honors referential and transactional integrity. Default is Transactional Integrity for Major Relation Type.
Email	An email address to send notifications to when the plan execution completes. Click settings to configure the email address, subject, and text of the email.
Target Load Type	Choose Normal or Bulk. If you select Normal, the PowerCenter Integration Service loads targets normally. Choose Normal mode if the mapping contains an Update Strategy transformation. Choose Bulk to improve session performance. Bulk load limits the ability to recover because no database logging occurs. Default is Normal.

Advanced Options	Description
Target Schema Name	Table name prefix. Specify the target schema name if the target is in a different schema. Choose to enter a value or parameter. If you choose to enter a parameter, select the parameter name from the list. If the target is Cassandra, you must specify the target schema name. If the target is MongoDB, you must not specify the target schema name. TDM uses the data source name that you specify in the ODBC Data Source Administrator when you configure the MongoDB data source. TDM does not override the data source name with the target schema name at the plan level. If you specify the target schema name for MongoDB, the workflow fails.
Source Schema Name	Table owner name. Specify the source schema name if the source is in a different schema. Choose to enter a value or parameter. Enter the schema name in uppercase unless the schema name is case sensitive in the database. If the schema name is case sensitive in the database, enter the name as in the database. If you choose to enter a parameter, select the parameter name from the list. If the source is Cassandra, you must specify the source schema name. If the source is MongoDB, you must not specify the source schema name. TDM uses the data source name that you specify in the ODBC Data Source Administrator when you configure the MongoDB data source. TDM does not override the data source name with the source schema name at the plan level. If you specify the source schema name for MongoDB, the workflow fails.
Batch Update	Enables batch update to update target rows in a batch mode. Improves performance. Use with inplace masking. **Note:** You cannot perform batch updates if you use connection variables because you cannot use a parameter for the staging connection.
Batch Size	The number of target rows that you want to move at a time. This option is available when you select the Batch Update as Yes. Use with inplace masking. **Note:** If a primary key or a unique constraint column contains a masking rule assignment, the batch size must be greater than or equal to the total number of records.
Recursive Executor Batch Update Size	The number of rows that you want to update at a time in a data subset operation on tables with cyclic or tending to cyclic constraints. Required when you choose to use batch updates for recursive executor.
Use Oracle ROWID in Source	Determines whether to use the ROWID from Oracle sources as a column. The ROWID indicates the physical location of a row in the database. Enabled when the Oracle source and target connections are the same.
Join parent and child tables using	Choose how to join the parent and the child tables in the SQL. Choose one of the following options: - IN clause. At least one value in the parent table must match a value in the child table. - EXISTS clause. The join must return at least one row.
Enable special characters in metadata	Select whether to allow special characters in plan metadata. Choose one of the following options: - Yes. Select yes to allow special characters in metadata. - No. Select no to disable special characters in metadata. Default is No.
Reset Recovery prefix	Required if you are running the plan to recover records deleted during a failed reset operation. The backup recovery prefix that you entered during the reset operation.

Advanced Options	Description
Generate Mappings With Source Data type	Select Yes if the data source was created with the ODBC connection type and if the source table or column names contain special characters. Select Yes when you want to generate a mapping with the actual source data type if ODBC does not support a data type. Applicable for databases that do not have native drivers support on Linux.
Shared Storage Table	Enables sharing of the storage table between Data Masking instances. Enable Shared Storage Table when two Data Masking instances use the same dictionary column for the database connection, seed value, and locale. You can also enable the shared storage table when two ports in the same Data Masking instance use the same dictionary column for the connection, seed, and locale. Disable the shared storage table when Data Masking instances or ports do not share the dictionary column. Default is disabled.
DTM Buffer Size	The Data Transformation Manager (DTM) buffer size specifies the amount of buffer memory that the PowerCenter Integration Service uses when the DTM processes a session.
Default Buffer Block Size	The buffer block size specifies the amount of buffer memory used to move a block of data from the source to the target.
SSN High Group File	The path to the High Group List directory that contains the Social Security numbers issued by the Social Security Administration.
DMO Cache Directory	The location of the dictionary cache. You must have write permissions for the directory. Default is $PMCacheDir.
Cache Size	The size of the dictionary cache in main memory. Increase the memory size in order to improve performance. Minimum recommended size is 32 MB for 100,000 records. Default is 8 MB.
Storage Commit Interval	The number of rows to commit at a time to the storage table. Increase the value to increase performance. Configure the commit interval when you do not configure the shared storage table. Default is 100,000.
Pre 85 Timestamp Compatibility	Trims subseconds to maintain compatibility with versions prior to 8.5. The PowerCenter Integration Service converts the Oracle Timestamp data type to the Oracle Date data type. The PowerCenter Integration Service trims subsecond data for the following sources and targets: - Relational sources and targets - XML sources and targets

Log Settings

Configure the log properties in the **Log Settings** fields. The section is minimized by default. The options are populated with default data if available.

The following table describes the log settings:

Log Settings	Description
Save Workflow Logs	You can create workflow logs according to the following options: - By Runs. The PowerCenter Integration Service creates a designated number of workflow logs. Configure the number of workflow logs in the **Save Workflow Log for These Runs** option. The PowerCenter Integration Service does not archive binary logs. - By Time Stamp. The PowerCenter Integration Service creates a log for all workflows, appending a time stamp to each log. When you save workflow logs by time stamp, the PowerCenter Integration Service archives binary logs and workflow log files.
Save Workflow Logs For These Runs	Number of historical workflow logs you want the PowerCenter Integration Service to create. The PowerCenter Integration Service creates the number of historical logs you specify, plus the most recent workflow log.
Write Backward Compatible Workflow Log	Writes workflow logs to a text log file. Select this option if you want to create a log file in addition to the binary log for the log events.
Save Session Log By	You can create session logs according to the following options: - Session Runs. The PowerCenter Integration Service creates a designated number of session logs. Configure the number of session logs in the **Save Session Logs for These Runs** option. The PowerCenter Integration Service does not archive binary logs. - Session Time Stamp. The PowerCenter Integration Service creates a log for all sessions, appending a time stamp to each log. When you save session logs by time stamp, the PowerCenter Integration Service archives binary logs and workflow log files.
Save Session Log For These Runs	Number of historical session logs you want the PowerCenter Integration Service to save. The PowerCenter Integration Service creates the number of historical logs you specify, plus the most recent session log.
Session Log file Maximum Size	The maximum number of megabytes for a log file. Configure a maximum size to enable log file rollover by file size. When the log file reaches the maximum size, the PowerCenter Integration Service creates a new log file. If you set the size to zero the session log file size has no limit. Configure this option for real-time sessions that generate large session logs. The PowerCenter Integration Service writes the session logs to multiple files. Each file is a partial log file. Default is zero.
Session Log file Maximum Time Period	The maximum number of hours that the PowerCenter Integration Service writes to a session log. Configure the maximum time period to enable log file rollover by time. When the period is over, the PowerCenter Integration service creates another log file. Configure this option for real-time sessions that generate large session logs. The PowerCenter Integration Service writes the session logs to multiple files. Each file is a partial log file. Default is zero.

Log Settings	Description
Maximum Partial Session Log Files	Maximum number of partial session log files to save. The PowerCenter Integration Service overwrites the oldest partial log file if the number of log files has reached the limit. Configure this option in conjunction with the maximum time period or maximum file size option. You must configure one of these options to enable session log rollover. If you configure a maximum of zero, the number of session log files is unlimited. Default is one.
Write Backward Compatible Session Log	Writes session logs to a text log file. Select this option if you want to create a log file in addition to the binary log for the log events.

Masking Components

Masking components are the policies and masking rules in a data masking operation. When you create a plan for a data masking operation, select the data masking components that you want to include in the workflow.

When you run a data masking operation, you can restrict the masking operation to some of the masking rules instead of applying all of the masking rules. You can choose the policies and masking rules that you want to test.

The **Masking Components** dialog box shows the policies and rules in the project. Select the policies and rules that you want to apply in the masking operation. The PowerCenter Integration Service performs masking with the components that you select. You can disable and enable masking for specific columns in a workflow run when you configure the plan criteria.

Subset Components

Subset components are the entities, groups, and templates or XSD files in a data subset operation. When you create a plan for a data subset operation, select the subset components that you want to include in the workflow.

When you run a data subset operation on a relational or flat file source, you can restrict the subset operation to specific entities in the project. You can choose the entities, groups, and templates that you want to include.

When you run a data subset operation on an XML data source, you can select the XSD files that you want to include in the operation.

The **Add Subset Components** dialog box shows the entities, groups, templates, or XSD files in the project. Select the components that you want to apply in the data subset operation. The PowerCenter Integration Service performs the data subset operation with the components that you select.

Generation Components

Generation components are the entities and tables in a data generation operation. If you select a target XML metadata, you must select the XSD file component in the data generation plan.

When you create a plan for a data generation operation, select the data generation components that you want to include in the workflow. To run a data generation operation, you can restrict the generation operation to some of the generation rules instead of applying all the generation rules. You can choose the generation rules that you want to test. When you configure the plan criteria, you can enter the number of records that you want to generate.

The **Generation Components** dialog box shows the entities and tables in the project. Select the components that you want to apply in the data generation operation.

Component Criteria

You can define filter criteria for data subset components and for data masking components in a plan. Define the filter criteria for each column you need to filter.

You can define expressions to limit data values for specific columns in a data subset operation. You can also limit the value to a percentage of rows, an interval of rows, or to an absolute value. You can disable and enable data masking for a column.

Define filter criteria for a column in the **Plan Components** panel. Choose the data masking or data subset component to filter.

Filtering Data Subset Components

You can define filter criteria for data subset components at the column level.

1. To create a data subset filter for a column, click the data subset component in the **Plan Components** panel.
2. Select a column from the **Criteria** panel to apply the filter to.

   Test Data Manager shows existing filter criteria. You can change the filter criteria or add new criteria.
3. Choose a method to filter the column data:

Option	Description
Custom	Create a basic expression or an advanced expression to filter the data.
Limit	Limit the subset results by percentage, by absolute value, or by an interval of rows to write to the target.

4. If you choose the **Custom** method, select one of the following options to define a filter:

Option	Description
Basic	Define a simple expression that includes just an operator and operand. You can use parameters in the expression.
Advanced	Define an expression in the **Expression Editor**. You can include multiple columns and functions in the expression. You can use parameters in the expression.

Disabling Masking for a Column

You can disable or enable masking for a column in the plan.

1. Click a masking component in the **Plan Components** panel.
2. Select a column from the **Criteria** panel.
3. To disable masking for the column, click **OFF**.
4. To enable masking for the column, click **ON**.

Source Settings

The source level properties are a subset of plan settings. You can change the settings for all the tables in the source or you can change the settings for each table in the source.

If a property is not available at the data source level, the property value at the plan level takes precedence. A lookup connection appears in the settings if there is a mapplet rule that is associated with a column in the data source.

To change settings at the source or table level, choose the **Source Settings** tab when you edit the plan.

Connection Properties

The following table describes connection options that you can configure at the data source level and the table level:

Connection Options	Description
Source Connection	A connection to the source database. Choose variable or relational from the list. Select the variable name or select a source connection from the list. You cannot use a Hadoop HDFS connection as a variable when you configure a source or a target.
Source Connection Directory Path	The path to the flat file source directory. Appears if flat file sources are present.
Source Filename	The name of the source file. Required if the source is a flat file. Default is <name of selected source>.dat.

Connection Options	Description
Target Connection	A connection to the target database or the target file. Choose variable, relational, or flat file from the list. Select a target connection or variable name from the list. If you enter the target connection as a variable, ensure that you define the scope of the variable as **Global** or **Integration Service** in the parameter file. In a multinode setup, if you use parameters to enter the connection information, the parameter file must be present on the node on which the Test Data Manager Service runs. The file must be in the same directory path on both nodes.
Target JDBC Connection	Appears if you select an ODBC target connection type. The connection that contains the JDBC connection string of the ODBC target database. Required if the target connection type is ODBC and either of the following statements is true: - You want to disable and enable constraints in the plan. - The source connection type is ODBC and you want to generate mappings with the source data type. Optional if the target is ODBC and you want to perform simple data movement operations.
Output Filename	The name of the target file. Required if the target is a flat file. Default is <name of selected source>.out.
File Encoding	The file encoding type. Required if the target is a flat file. If the source is a flat file, the default type is what the flat file contains. If the source is relational, default is MS Windows Latin 1 (ANSI), superset of Latin1. You can select the type of flat file encoding that you want in the target.
File Format	The format of a flat file. Required if the target is a flat file. You can choose one of the following options: - Fixed Width. The width of the columns is fixed. You cannot specify a column delimiter. - Delimited. You can limit the width of the columns. Specify a column delimiter. Default is Delimited.
Column Delimiter	A character that separates columns of data. Required if the source is a flat file and if the file format is delimited. Default is a comma (,).
Row Separator	A character that separates rows of data. Required if the source is a flat file. Default is \012 LF (\n).

Connection Options	Description

Optional Quotes	Select No Quotes, Single Quote, or Double Quotes. If you select a quote character, the Integration Service ignores delimiter characters within the quote characters. Therefore, the Integration Service uses quote characters to escape the delimiter.
	For example, a source file uses a comma as a delimiter and contains the following row:
	`342-3849, 'Smith, Jenna', 'Rockville, MD', 6.`
	If you select the optional single quote character, the Integration Service ignores the commas within the quotes and reads the row as four fields.
	If you do not select the optional single quote, the Integration Service reads six separate fields.
	When the Integration Service reads two optional quote characters within a quoted string, it treats them as one quote character. For example, the Integration Service reads the following quoted string as
	`I'm going tomorrow:`
	`2353, 'I''m going tomorrow', MD`
	Additionally, if you select an optional quote character, the Integration Service reads a string as a quoted string if the quote character is the first character of the field.
	Note: You can improve session performance if the source file does not contain quotes or escape characters.

Include Headers	Optional. You can choose to include the headers in the target flat file. Default is Yes.
Source File Name	The name of the IMS source file or VSAM source file. If the source is a Hierarchical IMS file, enter the name of the source unload file. If the source is a Hierarchical VSAM/SEQ file, enter the name of the source file. Required if the source connection is PXW_NRDB Batch.
Target File Name	The name of the IMS target file or VSAM target file. If the target is a Hierarchical IMS file, enter the name of the target unload file. If the target is a Hierarchical VSAM/SEQ file, enter the name of the target file. The target file name must be different from the source file name. Required if the target connection is PXW_NRDB Batch.
Teradata ODBC Connection	Name of the Teradata ODBC connection. Required if the target connection is Teradata PT. Select a connection from the list.
Lookup Connection	Connection to a database that contains lookup tables. Required if the plan has a mapplet rule which includes a transformation. Choose relational or flat file from the list. Select a lookup connection from the list.
Dictionary Connection	A connection to the database that contains the dictionary table. The dictionary connection option does not appear unless the project contains a rule that requires a relational dictionary.
Email Dictionary Connection	A connection to the database that contains the email dictionary table. The email dictionary connection option does not appear unless the project contains a masking rule that requires a relational email dictionary.

Test Tool Integration Properties

The following table describes the test tool integration properties that you can configure at the data source level:

Test Tool Integration Options	Description
Attach to Test Tool	Select this option to copy the flat file target results to a location in the HP ALM server integrated with TDM. You must run the workflow from Test Data Manager. The test tool integration job does not run when you run the workflow from the PowerCenter client.
Project	Required. Browse to select the project in the HP ALM server where you want to store the flat file results.

Target Properties

The following table describes target options that you can configure at the data source level and at the source level:

Other Property Options	Description
Truncate Table	Truncates the table before loading it. Default is disabled.

Update Strategy Properties

The following table describes strategy options that you can configure at the data source level:

Update Strategy Option	Description
Treat source rows as	Indicates how the PowerCenter Integration Service treats source rows. Choose Insert or Update. If an imported PowerCenter mapplet contains a transformation that is configured to set the update strategy, the default option is Insert.
Update as	Determines whether the PowerCenter Integration Service updates or inserts rows. Choose one of the following options: - Update. The PowerCenter Integration Service updates all rows flagged for update. - Insert. The PowerCenter Integration Service inserts all rows flagged for update. - Update else Insert. The PowerCenter Integration Service updates rows flagged for update if they exist in the target, and inserts remaining rows marked for insert.

Error and Recovery Properties

The following table describes connection options that you can configure at the data source level:

Error and Recovery Options	Description
Commit Type	Choose a source-based or target-based commit. By default, the PowerCenter Data Integration Service performs a target-based commit.
Commit Interval	The number of rows to process before a commit. Default is 10,000.

Source and Target Properties

You can override the name of the table owner if you want to select a table from a different schema.

The following table describes source and target options that you can configure at the data source level:

Source and Target Options	Description
Imported Source Owner Name	The default name of the source owner.
Runtime Source Owner Name	The name of the table owner from which you want the source data. Specify the owner name of the source table if you need the table from a different schema. Enter the schema name in uppercase unless the schema name is case sensitive in the database. If the schema name is case sensitive in the database, enter the name as in the database.
Target Prefix Name	The name prefix of the target table. Specify the target schema name if the target is in a different schema.

Advanced Properties

The following table describes advanced options that you can override at the data source level and at the table level:

Advanced Options	Description
Datetime Format String	Date-time format defined in the session properties. You can enter seconds, milliseconds, microseconds, or nanoseconds. - Seconds. MM/DD/YYYY HH24:MI:SS - Milliseconds. MM/DD/YYYY HH24:MI:SS.MS - Microseconds. MM/DD/YYYY HH24:MI:SS.US - Nanoseconds. MM/DD/YYYY HH24:MI:SS.NS Default is microseconds.
Target Load Type	Choose Normal or Bulk. If you select Normal, the PowerCenter Integration Service loads targets normally. Choose Normal mode if the mapping contains an Update Strategy transformation. Choose Bulk to improve session performance. Bulk load limits the ability to recover because no database logging occurs. Default is Normal.
Target Schema Name	Table name prefix. Specify the target schema name if the target is in a different schema. Choose to enter a value or parameter. If you choose to enter a parameter, select the parameter name from the list. If the target is Cassandra, you must specify the target schema name. If the target is MongoDB, you must not specify the target schema name.
Source Schema Name	Table owner name. Specify the source schema name if the source is in a different schema. Choose to enter a value or parameter. Enter the schema name in uppercase unless the schema name is case sensitive in the database. If the schema name is case sensitive in the database, enter the name as in the database. If you choose to enter a parameter, select the parameter name from the list. If the source is Cassandra, you must specify the source schema name. If the source is MongoDB, you must not specify the source schema name.
Source Pre SQL	SQL query executed before the plan is run. Required if you need to perform an operation before you run the plan. Enter the complete SQL statement.
Source Post SQL	SQL query executed after the plan is run. Required if you need to perform an operation after you run the plan. Enter the complete SQL statement.
Batch Size	The number of target rows that you want to move at a time. This option is available when you select the Batch Update as Yes. Use with inplace masking. **Note:** The batch size must be greater than the total number of records.
Use Oracle ROWID in Source	Determines whether to use the ROWID from Oracle sources as a column. The ROWID indicates the physical location of a row in the database. Enabled when the Oracle source and target connections are the same.
Performance Hint	Optimizes the SQL query after the plan is run. Enter a performance hint value only for Oracle databases.

.

Partitioning Properties

You must specify partitioning details if the tables in the source database contain partitions.

The following table describes partitioning options that you can configure at the table level:

Partitioning Property	Description
Enable Partitioning	Select to enable partitioning of output tables.
Partition Type	Select the type of partitioning: - Pass Through. Use pass through partitioning to increase data throughput without increasing the number of partitions. - Key Range. Use key range when the source data tables are partitioned by key range. - Database Partitioning. Use database partitioning for Oracle or IBM DB2 for Linux, UNIX and Windows sources and targets.
Number of partitions	The number of partitions in the source data. Enter the number of partitions and click **Select** to enter partition information for the target data.

XSD Data Source Connection Settings

You can override the plan settings and enter different source and target file directories in a plan that contains XSD data sources. The settings apply to all files in the directory. You can override the data source connection settings for individual files in the directory.

The following table describes the connection options that you can configure at the data source level and at the file level:

Connection Options	Description
Source File Directory	The path to the XML source file directory. The source file must reside on the server on which the PowerCenter Integration Service runs.
Target File Directory	The path to the XML target file directory.
Source Type	Indicates whether the source file contains the source data or whether the source file contains a list of files with the same file properties. Use to configure multiple file sources with a file list. - Direct. The source file contains the source data. - Indirect. The source file is an index file. The index file contains a list of file names. The PowerCenter Integration Service finds the index file and reads each listed file when it runs the session. The index file must have an empty line at the end of the file. Default is Direct.

Connection Options	Description
Source File Name	The file name or file name and path. Optionally, use the $InputFileName session parameter for the file name. If you specify both the directory and file name in the **Source File Directory** field, clear this field. The PowerCenter Integration Service concatenates this field with the **Source File Directory** field when it runs the session. For example, if you have "C:\XMLdata\" in the **Source File Directory** field, then enter "filename.xml" in the Source File Name field. When the PowerCenter Integration Service begins the session, it looks for "C:\XMLdata\filename.xml".
Output File Name	The file name, or file name and path. Optionally, use the $OutputFileName session parameter for the file name. If you specify both the directory and file name in the **Output File Directory** field, clear this field. The PowerCenter Integration Service concatenates this field with the **Output File Directory** field when it runs the session.

XSD Data Source Advanced Settings

You can override the plan settings for XSD sources in a plan that contains XSD data sources. The settings apply to all files in the directory. You can override the data source connection settings for individual files in the directory.

The following table describes the advanced options that you can configure at the file level:

Advanced Options	Description
XML date-time format	Date-time format defined in the session properties. Choose one of the following values: - Local Time. The time according to the PowerCenter Integration Service server time zone. - Local Time with Time Zone. The difference in hours between the PowerCenter Integration Service time zone and Greenwich Mean Time. - UTC. Greenwich Mean Time. Default is Local Time.
Treat empty content as NULL	Treat empty XML components as null. By default, the PowerCenter Integration Service does not output element tags for null values. The PowerCenter Integration Service outputs tags for empty content. Default is No.
Format Output	Format the XML target file so the XML elements and attributes indent. If you do not select Format Output, each line of the XML file starts in the same position. Default is No.
NULL content representation	Choose how to represent null content in the target. Default is No tag.
Empty String content representation	Choose how to represent empty string content in the target. Default is Tag with empty content.
NULL attribute representation	Choose how to represent null attributes. Default is No attribute.
Empty String attribute representation	Choose how to represent empty string attributes in the target. Default is Attribute with empty string.

Plan Management

After you create a plan you can edit it. You can copy the plan, export it to an XML file, or delete the plan. You can import a plan that you created in another TDM repository and exported.

Creating a Data Masking and Data Subset Plan

When you create a plan, add components to it to define its operations. You can combine a data subset and a data masking operation in the same plan, or create separate plans. To perform a data generation operation, you must create a separate plan. Add entities, groups, and templates to complete data subset operations. Add rules and policies to plans to perform data masking operations.

1. Open a project and click **Execute** to view the project plans.
2. Click **Actions** > **New**.
3. In the **New Plan** dialog box, enter a name and optional description for the plan.
4. To add a data masking operation to the plan, click **Add Masking Components**.
5. Select the policies and rules to add to the plan. Click **Next**.
6. To add a data subset operation to the plan, click **Add Subset Components**.
7. Select the entities, groups, and templates, or XSD files to add to the plan. Click **Next**.

   You can assign criteria for masking columns and generating subsets.
8. To skip adding a data generation component, click **Next**.
9. To skip masking a rule, select the check box for the rule and click **Off**.
10. To filter subset components, select the component and choose to enter a basic or advanced expression.

    Test Data Manager shows any existing filter criteria. You can change the criteria. You can use parameters in an advanced expression. You cannot override subset criteria for an XSD subset component.
11. To limit the subset results, click **Limit** and choose to limit by percentage, by absolute value, or by defining an interval of rows to create. Click **Next**.
12. Configure the connections and other properties. If the target is a flat file, you can configure test tool integration properties. If you select the test data warehouse as the target connection, you can enter the data set properties. If the source is an XSD source, you cannot add other types of sources to the plan. Click **Next**.
13. To override plan settings for a data source, select the data source and click **Override Plan Settings** and enter the properties.
14. To override data source settings for a table, select the table and click **Override Data Source Settings** and enter the properties.
15. Click **Finish**.

    The plan appears in the project.

Related Topics:

- "Generating a Workflow" on page 293
- "Advanced Masking Rules" on page 129
- "Advanced Masking" on page 148

Creating a Data Generation Plan

To perform a data generation operation, you create a plan and add data generation components to the plan. When you add data generation components, you cannot add data subset or data masking components to the same plan.

1. Open a project and click **Execute**.
2. Click **New Plan**.
3. In the **New Plan** dialog box, enter a name and optional description for the plan.
4. Click **Next**.
5. To skip adding the masking components, click **Next**.
6. To skip adding the subset components, click **Next**.
7. To add the generation components to the plan, click **Add Generation Components**.
8. Select the entities and the tables to add to the plan. If you want to generate test data in an XML file, select the XSD file to add to the plan.
9. Click **OK**.

   The entities and tables appear in the list.
10. Click **Next**.
11. To view or filter the table names, select the table.
12. To specify generation criteria for an entity, select the entity and enter the values.
13. Click **Next**.
14. To configure generation plan settings, enter the number of records that you want to generate.
15. Configure connection and target properties. If the target is a flat file, you can configure test tool integration properties. If you select the test data warehouse as the target connection, you can enter the data set properties.
16. To make changes to data generation, error recovery, and advanced settings, you can edit the settings.
17. Click **Next**.
18. Click **Next**.
19. To override plan settings, click **Override Plan Settings** and enter the properties.
20. To override table settings, click **Override Data Source Settings** and enter the properties.
21. Click **Finish**.

    The generation plan appears in the project.

Creating an Ad Hoc Generation Plan

You can create an ad hoc generation plan for a column.

1. Open a project and click **Define** > **Data Generation**.
2. Select a column from the table for which you want to create an ad hoc generation plan.
3. Click **Actions** > **Generate and Execute**.

   The **Generate and Execute Plan** dialog box appears.
4. Enter a name for the plan or keep the default plan name that appears on the screen.
5. Select the Integration Service.
6. Enter the number of records that you want to generate.

7. Configure connections and target properties.
8. To make changes to data generation, error and recovery, and advanced settings, edit the settings.
9. Click **OK**.

   A **Create and Execute** dialog box appears.
10. To run the plan, click **Yes**.

Copying a Plan

Copy a plan to create another plan with similar components. Create a copy of a plan and edit the copy.

1. Open a project and click **Execute** to view the project plans.
2. Click the plan **Description** or **Status** field to select a plan.

   Do not open the plan.
3. Click **Actions** > **Duplicate**.
4. Optionally, enter a plan name and a description.

   The default name is `Copy of <original name>`.

Exporting a Plan

You can export a group to an XML file and import the XML file to another TDM repository.

1. Open a project and click **Execute** to view the project plans.
2. Click the plan **Description** or **Status** field to select a plan.

   Do not open the plan.
3. Click **Actions** > **Export**.
4. Choose to save the file.
5. Enter the XML file name and the path of the file.

   The default name is a string that contains "Plan_" and the current date and the time.

Importing a Plan

You can import a plan from an XML file that was exported from another TDM repository.

1. To open the **Projects** view, click **Projects**.
2. Click the project description to select a project to import the plan into.

   Do not open the project.
3. Click **Actions** > **Import**.
4. Browse for the XML file that contains the group to import.

   The XML file has a default name similar to `Plan_130315081854.xml`.
5. Click **Finish** to import the plan.

Deleting a Plan

You can delete plans. When you delete the plan, you delete the workflow for the plan.

1. Open a project and click **Execute** to view the project plans.
2. Click the plan **Description** or **Status** field to select a plan.

   Do not open the plan.
3. Click **Actions** > **Delete**.
4. Click **Yes** to confirm the delete.

Workflow Generation

Generate a workflow from a plan, and then run the workflow to perform data subset, data masking, and data generation operations.

You can generate and start a workflow in a single step.

When you generate a workflow, the PowerCenter Repository Service generates mappings for the workflow. When you start a workflow, the PowerCenter Integration Service performs the data masking, data subset, and data generation tasks defined in the plan. You can select the PowerCenter Integration Service to run the workflow.

You can view the status of generated workflows or the load configuration on the **Monitor** view.

Generating a Workflow

After you create a plan, generate a workflow. The PowerCenter Repository Service generates the mappings to include in the workflow.

1. In a project click **Execute** to access the plans in the project.
2. Select a plan from the list.
3. Click **Actions** > **Generate Workflow**.
4. Choose whether to run it immediately or to generate the workflow later.

Related Topics:

- "Creating a Data Masking and Data Subset Plan" on page 290
- "Advanced Masking Rules" on page 129
- "Advanced Masking" on page 148

Export to HP ALM

You can export flat file results of a plan in TDM to an HP ALM server. You must enable and configure test tool integration before you can export the results.

Enter the test tool integration settings when you define the plan settings. TDM runs a separate job in the same workflow to copy the flat file to the HP ALM server in addition to the TDM server. The test tool integration job creates a test resource folder in the HP ALM server and appends the plan name and the job ID to the folder name.

You must run the workflow from Test Data Manager. The test tool integration job does not run if you run the workflow from the PowerCenter client.

Note: You cannot export a flat file from TDM if you enable version control on the HP ALM server. You must disable version control and then run the TDM workflow.

Workflow View

The Workflow view contains a list of the workflows that you generate from a plan. You can view details about the workflows and sessions that you generated from the plan.

The Workflow Details panel contains the list of workflows in the plan and the properties panel contains the workflow properties and session names.

The **Workflow** view has a **Workflow Details** panel and a **Properties** panel. The **Workflow Details** panel contains the list of workflows in the plan. The Properties panel contains properties for a workflow and the session names in the workflow.

Properties Panel

The **Properties** panel in the **Workflow** view contains a list of the workflows generated from a plan.

You can view the following information about generated workflows:

Property	Description
Workflow Name	A string that includes the project name, plan name, connection, and table name.
User	The name of the user that generated the workflow.
Generation	Date and time that the workflow was generated.

Details Panel

The **Details** panel shows the **Workflow** details and the **Session** details for a workflow that you select in the **Workflows** view.

The **Workflow** details shows the summary information about a workflow. When you select a workflow in the **Workflow** view, the workflow details appear. You can view the workflow name, the user, and the project name.

The **Session** details panel appears next to the **Workflow** details panel. The panel shows each session name and the number of tables that the session processes for a workflow.

Parameter Files in Test Data Manager

Create a parameter file to view a list of all parameters used in a plan. You can use the parameter file to include or exclude parameters and to edit parameter values when you run a workflow.

When you create a parameter file from a plan, the file includes all parameters in the plan with the default values that you add during parameter creation.

The parameter file contains ignore flags for criteria parameters. By default the flags are set to "N". This means that by default a workflow that uses the parameter file includes the criteria parameters. To ignore a criteria parameter in a workflow, set the ignore flag to "Y" and save the file before you use it in a workflow.

Change parameter values each time you run a workflow to create test data on demand. For example, you want to create data subsets with different filter criteria values although the criteria is the same. Use a criteria parameter in the entity and the plan. You can then change the value of the criteria parameter to change the output each time you run the workflow. You can create different test data output when you choose specific criteria parameters to use each time you run the workflow.

Use connection parameters and owner parameters in a plan and change the parameter values in the parameter file to run a workflow on different source data each time.

Creating a Parameter File

Create a parameter file to view the parameters in a plan and edit the parameter values before you run a workflow.

1. Open a project and click **Execute** to view the project plans.
2. Select the required plan.

   You can also open the required plan to view the **Properties** tab.
3. Click **Actions** > **Create Parameter File**.

   The file downloads to the default download location. You can use a text editor to view the parameter file from the location provided.
4. If you configure the browser to prompt for a download location, you must enter a file path and file name.

   Click **Save As** to create and download the parameter file.
5. Optional. To change parameter values in the file, enter the required values. To ignore criteria parameters, set the ignore flag to Y. Save the changes.

   Changes to parameters that you cannot override do not take effect.

To use the parameter values, select the file as the parameter file when you run the workflow.

Executing a Workflow

After you generate a workflow, you can run the workflow to run sessions from the PowerCenter mappings. If you generated multiple workflows for the same plan, you can run each workflow separately.

You can start a workflow from the **Plan | Properties** view or the **Plan | Workflow** view.

1. To generate and run a workflow in a single step, in the **Plan | Properties** page, click **Actions** > **Generate and Execute**.
2. You can generate and run a workflow in separate steps. If the plan you selected contains one workflow, click **Actions** > **Execute Workflow** in either view.
3. If the plan you selected contains multiple workflows, click the **Workflow** view.
   a. Select the workflow that you want to start.
   b. Click **Actions** > **Execute Workflow**.
4. If the plan contains parameters, you must select the parameter values to use in the workflow.

- Default. The workflow uses the parameter values in the parameter file that you specify in the **Plan Settings** page of the plan.
- Use Parameter File. The workflow uses parameter values in a parameter file that you choose. Click **Browse** and select the required parameter file.
- Add Parameter Values. The workflow uses parameters that you create in the project. A list of parameters included in the plan appears.
  - You can choose to use the default values.
  - You can choose to edit the default values. To edit a parameter value, click the required row and enter the value in the value column. Click the **Save** button.
  - Criteria parameters do not appear in the list by default. To include criteria parameters, click the **Select Parameter** button to view and select parameters. Parameters that you select appear in the list of parameters. You can then edit the parameter value or use the default value.

    Note: A workflow does not include parameters that do not appear in the list. Select and add the required criteria parameters to the list to ensure that the workflow includes the criteria parameters.

  Note: If you edit the value of a parameter that you cannot override, the change does not take effect. The workflow uses the default value and generates an error in the logs.

  Select the required option and click **Save**.

5. If the target is the test data warehouse, you can choose to publish the data set to the self-service portal. Test Data Manager publishes the data set to the self-service portal as a data pack.
   a. To publish the data set to the self-service portal, expand the **Publish to the Self-Service Portal** section.
   b. Enter a name and optional description for the data pack.
   c. Optional. Add tags to the data pack. Use a comma to separate tags.

      To delete a tag that you enter, click the **X** icon that appears with the tag. To use the keyboard to delete a tag, press **Tab** or **Shift+Tab** to select the tag and then press **Delete** from the keyboard.
   d. Optional. To add users or user groups for the data pack in the self-service portal, click the **Select Users** button. The **Select Users for the Data Pack** dialog box appears.
   e. Optionally, you can filter and search for the list of users or user groups to add.
   f. Optional. Select the users or user groups to add to the data pack and click **Select**.

      Note: TDM adds the current user and administrator users to the data pack by default. You can add other users if required.
6. Click the **Generate and Execute** button to run the workflow.
7. View the workflow status in the **Workflow Execution** view.

Note: If you run the workflow with the **Disable Constraints** option selected, and TDM fails to enable the constraints after loading the data, the workflow fails. You must then manually enable the constraints in the database and run the workflow again.

Workflow Executions View

The **Workflow Executions** view shows the current workflow that runs for the plan. You can also view the status of all the previous workflows. The list can contain workflow runs for workflows that no longer exist in

the plan. The order of the workflow is based on the start date and time. To copy flat files to an integrated test tool, the workflow runs a separate job.

To view the workflow log, you can click **Job ID**. To view the session log, you can click **Session ID** in the **Sessions** tab.

The following table describes the fields in the **Workflow Executions** view for each workflow run:

Field	Description
Job ID	The job number that identifies the job. If you click the Job ID, you can view the workflow log.
Name	The workflow string that includes the project name, plan name, connection, and table name.
Description	Describes the type of job that the workflow performs. The job type can be profiling, import, and workflow operations.
Status	The current status of the workflow. - In Queue. The PowerCenter Integration Service is waiting for resources before it starts the workflow. - Running. The workflow is running. - Success. The workflow finished successfully. - Error. The workflow did not complete due to errors.
Start Date/Time	The date and time that the workflow started.
End Date/Time	The date and time that the workflow ended.
User	The name of the user that started the workflow.

Workflow Tasks

You can stop, abort, and recover the workflow in the Workflow Executions view.

You can perform the following tasks on the Workflow Executions view:

Auto Refresh

Causes the view to refresh every ten seconds automatically. You can turn auto refresh on and off.

Abort

Stops a job immediately without waiting for a commit. You can abort all jobs, such as profiling and import. You cannot abort workflows.

Workflow Stop

Stops a workflow after the next commit occurs.

Workflow Abort

Stops a workflow immediately without waiting for a commit.

Workflow Recover

Recover a stopped workflow. If high availability recovery is enabled, the PowerCenter Integration Service restarts the workflow from the last saved checkpoint. Otherwise, the PowerCenter Integration Service recovers the workflow according to the recovery strategy that you set in the plan.

Workflow Properties Panel

The **Properties** panel shows summary information about a workflow that you select in the **Workflow Executions** view.

The **Properties** panel shows the same information as the workflow you select in the list. The properties also include generation date, the elapsed time, and the project name.

Workflow Sessions Tab

The **Sessions** tab lists the sessions that ran in the workflow that you select in the **Workflow Executions** view.

The following table describes the fields for each session in the **Sessions** tab:

Field	Description
Job ID	The job number that identifies the session. If you click the Job ID, you can view the session log.
Session Name	The session name is the same value as the workflow name, except the session name starts with "S" and ends with a sequential number. If you click the session name, you can view the session details.
Status	The current status of the session. - In Queue. The PowerCenter Integration Service is waiting for resources before it starts the session. - Running. The session is running. - Success. The session finished successfully. - Error. The session did not complete due to errors.
Source Rows Succeeded	The number of valid source rows read.
Source Rows Failed	The number of source rows in error.
Target Rows Succeeded	The number of rows written to the target.
Target Rows Failed	The number of rows that were not written to the target.
Start Date/Time	The date and time that the session started.
End Date/Time	The date and time that the session ended.
User	The name of the user that started the workflow.

Session Details

The **Session Details** dialog box contains a list of tables that a session processed. The **Session Details** dialog box appears when you click the session name in the **Workflow Execution Sessions** tab.

If the **Workflow Executions** view has refresh enabled, the session details refresh when the workflows refresh.

The following table describes the fields in the **Session Details** dialog box:

Field	Description
Tables	The names of the tables that the session processed.
Affected Rows	The number of rows that the session processed from the table.
Rejected Rows	The number of rows that the PowerCenter Integration Service did not process due to errors.
Throughput	The number of affected rows that the PowerCenter Integration Service did not process due to errors.
Last Error Message	The text of the last error message in the session. Click the link to view the complete message.

CHAPTER 14

Monitor

This chapter includes the following topics:

- Monitor Overview, 300
- Jobs, 301
- Monitor Tasks, 302
- Logs, 303
- Sessions, 304

Monitor Overview

In the **Monitor** view, you can monitor the status of jobs that you start in Test Data Manager. You can stop jobs from running and view job and session logs.

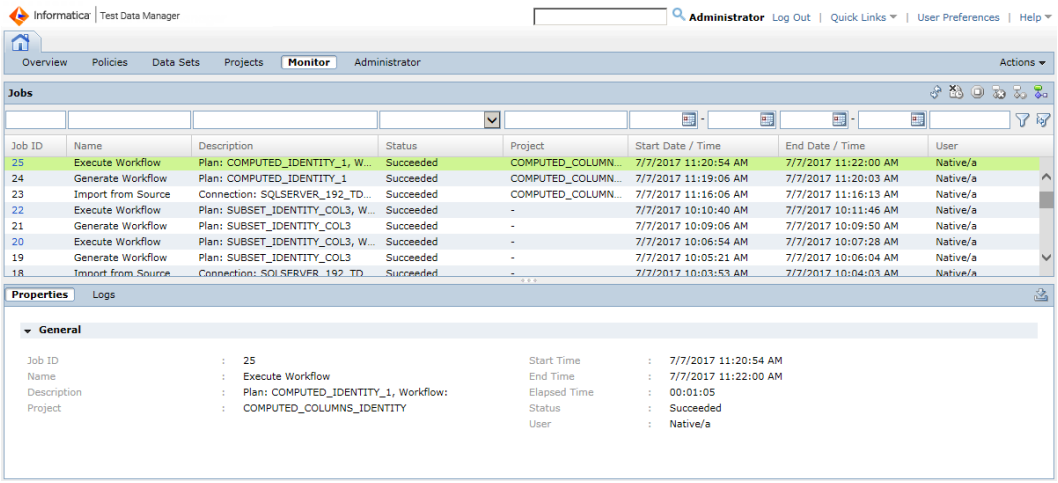
You can monitor jobs for all projects in the **Monitor** view. To monitor jobs for a single project, open the project and click **Monitor**.

You can monitor jobs for all data set versions in the **Monitor** view.

To monitor jobs for a specific data set version, open the data set version and click **Monitor**. You can view jobs run on the data set version from any machine that hosts a Test Data Manager Service in the domain.

You can sort, filter, and perform tasks on jobs in the **Monitor** view. Select a job in the **Monitor** view to view the job details in the **Properties** pane. You can also view the workflow and session logs for a job.

The following image shows the **Monitor** view:



Jobs

Check the status of a job and view the job details in the **Monitor** view.

You can view the following types of jobs in the **Monitor** view:

Import from Source

Imports source data from a source file.

Import from PowerCenter

Imports a source from the PowerCenter repository.

Profiling

Performs data discovery for primary keys, entities, and data domains.

Generate Workflow

Generates a workflow from a plan.

Execute Workflow

Runs a workflow for data subset, data masking, or data generation operations after you generate the workflow.

Workflow

Runs the data subset, data masking, or data generation operation. The execute workflow job might run multiple workflows.

Upload to test tool

Uploads all flat files created by a successful workflow to the integrated HP ALM server location.

Reset Data Set from Test Data Warehouse

Resets target data with a data set from the test data warehouse.

Data Coverage Analysis

Performs analysis in a data coverage task.

Fill Cell

Moves data from one cell to another cell in a data coverage task.

Session

Performs a task within the workflow. A workflow might have multiple sessions. Click on a workflow job ID to view the session details in another tab in the **Session** pane.

Job Details

You can sort and filter jobs by job details.

The **Monitor** view contains the following job details:

Job ID

The job ID number. TDM creates consecutive job ID numbers for each job. When you click on an Execute Workflow job, the workflow details open in a separate tab.

Name

The name or the type of job. You can view import, profiling, and workflow jobs.

Description

The name of the plan, profile, PowerCenter folder, or connection associated with the job.

Status

The status of the job. A job can have the following statuses:

- Error. The job did not run successfully. Click the job ID to view the job log file.
- In Queue. The job is in the queue to run.
- Running. The job is running.
- Success. The job ran successfully.
- Terminated. The job was terminated.

Project

The name of the project that contains the job. Project details are not visible in the **Monitor** view within a project.

Start Date / Time

The date and time the job started.

End Date / Time

The date and time the job ended.

User

The user that ran the job.

Monitor Tasks

You can perform tasks in the Monitor view based on the job that you view.

You can perform the following tasks in the Monitor view:

Auto Refresh

Refreshes the view every ten seconds. If you disable the auto refresh, click **Actions** > **Refresh** to manually refresh.

Unschedule

Removes a job that you scheduled to run at a later time. To permanently remove a job from a schedule, click **Actions** > **Unschedule**.

Note: If you upgrade to TDM 10.2.0, you cannot remove a job that you scheduled in TDM 10.1.0.

Abort

Stops a job immediately. You can abort all jobs except workflows.

Workflow Stop

Stops a workflow after the next commit occurs.

Workflow Abort

Stops a workflow immediately.

Workflow Recover

Recovers an aborted or stopped workflow. If high availability recovery is enabled, the PowerCenter Integration Service restarts the workflow from the last saved checkpoint. Otherwise, the PowerCenter Integration Service clears the state of operation and the recovery information before it restarts the workflow.

Logs

You can view logs to troubleshoot jobs. To view the logs, select a job and click the **Logs** tab. You can view the session, workflow, and console logs.

When a TDM job runs, the TDM server generates logs. The PowerCenter Integration Service generates the session and workflow logs. When a TDM job fails, you can view the logs to debug problems.

When a job is triggered, TDM always generates console logs. If the job logs are not available, you can view the console logs to check the messages.

When you click a job ID, you can view the following log details:

Date/ Time

The date and time the job ended.

Severity

The severity level of the log messages.

Description

The detailed description of the error message, the cause of the problem, and the solution.

Severity Levels

You can view the severity level of a log message and get a clear understanding of the problem level.

The log messages include the following severity levels:

Error

Indicates that the TDM server failed to perform an operation or respond to a request from a client application.

Warning

Indicates that the TDM server is performing an operation that might cause an error.

Info

Indicates that the TDM server is performing an operation that does not cause errors or problems.

Debug

Indicates TDM Server operations at a detailed level. The debug messages generally record the success or failure of server operations.

Trace

Indicates TDM Server operations at a more specific level than the debug logs. Trace messages are generally trace code paths.

Viewing the Log Messages

View the log messages to troubleshoot the problems when a TDM job fails. You can search and filter the log messages based on the dates, severity levels, and keywords in the error description.

1. Click **Monitor**.
2. Click a job ID and click the **Logs** tab.

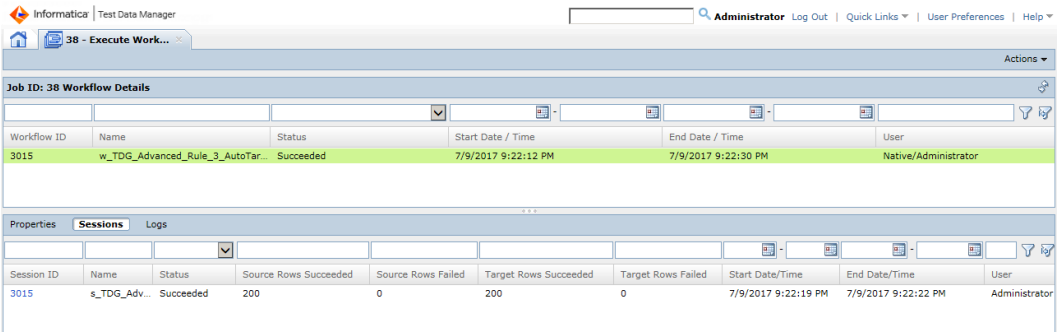
   A list of log messages appears.
3. Search and filter the logs from the list of log messages.
4. Select the log message and download the log file.
5. To download the log file, click **Download**.
6. To view console logs for each job, select a job ID and click **Actions** > **View Console Logs**.

Sessions

You can view the session logs after you run a workflow in Test Data Manager.

When you click a job ID within the execution workflow from the **Monitor** page, you can view all the session logs and workflow details in a separate tab. The **Sessions** tab lists the sessions that ran for the workflow.

The following image shows the session details for a workflow job:



The **Sessions** tab contains the following fields:

Session ID

The session number that identifies the session. If you click **Session ID**, you can view the processed tables, affected and rejected rows of the tables, error messages, and session logs.

Name

The name of the session that ran in the workflow.

Status

The current status of the session. The session can have the following statuses:

- In Queue. The PowerCenter Integration Service is waiting for resources before it starts the workflow.
- Running. The workflow is running.
- Success. The workflow completed successfully.
- Error. The workflow did not complete due to errors.

Source Rows Succeeded

The number of valid source rows read during the session.

Source Rows Failed

The number of source rows that failed due to errors.

Target Rows Succeeded

The number of rows written to the target.

Target Rows Failed

The number of rows that were not written to the target.

Start Date/Time

The date and time that the session started.

End Date/Time

The date and time that the session ended.

User

The name of the user that started the workflow.

CHAPTER 15

Reports

This chapter includes the following topics:

- Reports Overview, 306
- Audit Trail Report, 307
- Data Masking Report, 307
- Plan Audit Report, 307
- Plan Detail Report, 309
- Row Count Report, 309

Reports Overview

You can run a report from Test Data Manager to see detailed information about recent activities, a plan or a policy. Run the type of report that contains the information you want to see.

Test Data Manager contains activity, data masking, plan audit, plan detail, and row count reports. You can run a data masking report on a project. You can run a plan audit, plan detail, or a row count report on a plan.

When you run a report in Test Data Manager, the report opens in a new window. Make sure you do not have a pop-up blocker enabled so that the report window can open.

The following table describes the reports that you can run in Test Data Manager:

Report	Description
Audit Trail Report	A report that you run from the Overview view that lists all recent activities performed in Test Data Manager.
Data Masking Report	A report that you run on a project that lists all the table columns and the rules and policies that are assigned to the columns.
Plan Audit Report	A report that you run on a plan that contains detailed information about the policy assignments in the plan.
Plan Detail Report	A report that you run on a plan that lists policy assignments, entity details, groups, templates, and template components that are part of the plan.
Row Count	A report that you run on a plan that lists the tables in the plan and the number of rows in each table that the plan affects.

Audit Trail Report

An audit trail report lists the recent activities performed in Test Data Manager.

The report contains information on the type of activity, type of object, name of object, the user that performed the activity, and the date and time. You can apply filters to view specific activity records or to view activities for a specific date range.

An audit trail report does not consider permissions and privileges. A report that you generate includes all recent activities.

You can download the report in `.csv` file format.

Running an Audit Trail Report

Run an audit trail report to view all recent activities performed in Test Data Manager. You can download the report in CSV file format.

1. Log in to Test Data Manager.
2. Click the **Audit Trail Report** button in the **Recent Activities** tab.

   The Audit Trail Report window opens. The report lists all recent activities.
3. Optional. You can filter and view records that match specific requirements.
4. Optional. Click the **Download** button to download the report as a CSV file `Recent Activities.csv`.

Data Masking Report

A data masking report lists all the columns within a project and the rules and policies that you assign to the project.

Running the Data Masking Report

1. Click **Projects** to view a list of the projects in Test Data Manager.
2. Click a project in the list to open it.
3. Click **Define** > **Data Masking** > **Print Masking Report** to view the project plans.

   The data masking report opens in a new window.

Plan Audit Report

Run a plan audit report to view detailed information about the data masking components that took part in a plan execution.

You can generate a plan audit report to present to auditors or administrators that need comprehensive information about the policies and rules that took part in a plan execution.

You can create a plan audit report for a plan that has successfully completed at least one execution.

The following table lists the properties in a plan audit report:

Component	Properties
Policy Assignment	You can view the following policy assignment details: - Data Source - Table - Column - Data Type - Data Domain - Policy - Rule - Rule Type - Additional Details - Masking Properties - Exception Handling
Plan Execution	You can view the following plan execution details: - Source Connection - Target Connection - Created Date - Plan Start Date - Plan End Date - Total Rows Processed - Status of the Plan

Running a Plan Audit Report

1. Click **Projects** to view a list of the projects in Test Data Manager.
2. Click a project in the list to open it.
3. Click **Execute** to view the project jobs.
4. Click a plan.
5. Click **Actions** > **Plan Audit Report**.

   The row count report opens in a new window.

Plan Detail Report

A plan detail report lists the data subset and data masking components within a plan.

The following table lists the properties in a plan detail report:

Component	Detail
Masking Assignment	You can view the following policy assignment details: - Data Source - Table - Column - Data Type - Data Domain - Policy - Rule - Masking Properties
Entity	You can view the following entity details: - Name - Description - Filter Criteria
Group	You can view the following group details: - Name - Description
Template	You can view the following template details: - Name - Description - Filter Criteria
Template Component	You can view the following template component details: - Name - Type - Description - Component Name

Running the Plan Detail Report

1. To view a list of the projects, click **Projects**.
2. Click a project in the list to open it.
3. To view the project plans, click **Execute**.
4. Click a plan.
5. Click **Actions** > **Plan Detail Report**.

   The plan detail report opens in a new window.

Row Count Report

A row count report contains a list of the tables that took part in a plan execution and the number of rows that the plan affected.

Running the Row Count Report

1. To view a list of the projects, click **Projects**.
2. Click a project in the list to open it.
3. To view the project jobs, click **Monitor**.
4. Click a job that has an Execute Workflow job type.
5. Click **Actions** > **Row Count Report**.

   The row count report opens in a new window.

CHAPTER 16

ilmcmd

This chapter includes the following topics:

- ilmcmd Overview, 311
- Configuring ilmcmd, 312
- Running ilmcmd, 312
- Entering Options and Arguments, 313
- Syntax Notation, 313
- Delete, 314
- Export, 315
- Import, 316
- Search, 317
- Workflow, 319
- Reset, 320
- ListPlans, 322
- TDWPlanGenerate, 323
- TDWPlanExecute, 324
- TDWPlanGenExe, 324

ilmcmd Overview

ilmcmd is a command line program that you use to perform TDM tasks. You can use ilmcmd to complete a subset of the tasks that you can perform with Test Data Manager.

Use ilmcmd to perform the following tasks:

- Import XML files into the TDM repository.
- Export data subset, data masking, and data generation objects to XML files.
- Search for data subset and data masking objects.
- Validate policies, plans, and data subset objects.
- Delete data subset, data masking, and data generation objects.
- Generate and run workflows.
- Display the status of workflows.

You must have the required privileges to perform these tasks. For tasks related to projects, you must also have the required permission on the project.

You cannot use ilmcmd to search for data generation objects.

Configuring ilmcmd

When Test Data Manager runs on HTTPS, you must configure the ilmcmd before you can run ilmcmd commands.

Before you configure ilmcmd, you must import the browser certificate to the following location:

```
<Install directory>/TDM/utilities/ilmcli/conf
```

To generate a keystore, you must import the browser certificate. The keystore is necessary to run ilmcmd commands.

1. To set the password and generate a keystore, run the following command:

   ```
   keytool -import -file <Imported certificate> -keystore client.ks
   ```

2. Edit the userConfig.ilm file and add the keystore location for the javax.net.ssl.trustStore parameter. You can find the userConfig.ilm file in the following location:

   ```
   <Installation directory>/TDM/utilities/ilmcli/conf/userConfig.ilm
   ```

3. Add the password that you used to create the keystore for the javax.net.ssl.trustStorePassword parameter.

Running ilmcmd

Invoke ilmcmd from the command line. You can issue commands directly or from a script, batch file, or other program. On Windows, ilmcmd is a batch file with a .bat extension. On UNIX, ilmcmd is a shell script with no extension.

1. At the command line, switch to the directory where the ilmcmd executable is located.

   By default, ilmcmd is installed in the following directory:

   ```
   <installation directory>\utilities\ilmcli\bin
   ```

2. Enter `ilmcmd` followed by the command name and its required options and arguments.

   For example:

   ```
   ilmcmd -command_name [-option1] argument_1 [-option2] argument_2...
   ```

To view the command line syntax, enter the following command:

```
ilmcmd -h
```

Entering Options and Arguments

The ilmcmd command line program uses a set of options and arguments.

Use the following rules when you enter command options and arguments:

- To enter options, type a hyphen followed by the program syntax for the command.
- Enter options in any order.
- If an argument contains spaces, enclose the argument in double quotes.
- The first word after the option is the argument.
- Most options require arguments. You must separate options from arguments with a single space.
- Commands, options, and object names are case sensitive.

Syntax Notation

To use the ilmcmd command line program, review the syntax notation.

The following table describes the syntax notation for the ilmcmd command line program:

Convention	Description
-x	Option placed before an argument. This designates the parameter you enter. For example, to enter the user name for ilmcmd, type -un or -UserName followed by the user name.
<x>	Required option. If you omit a required option, the command line program returns an error message.
<x | y>	Select between required options. If you omit a required option, the command line program returns an error message. If a pipe symbol (|) separates options, specify exactly one option. If options are not separated by pipe symbols, specify all the options.
[x]	Optional option. The command runs whether or not you enter these options.
[x | y]	Select between optional options. For example, you can display help for all ilmcmd commands by using the -h or -help option. `[-h|-help]` The command runs whether or not you enter the optional parameter.
< <x | y> | <a | b> **>**	When a set contains subsets, the superset is indicated with bold brackets (**< >**). Subsets are separated with the bold pipe symbol (|).

Delete

Deletes objects from the TDM repository. To delete an object, specify the name of the object.

When you delete an object by name, you specify the object type and the location of the object in the repository. You can delete the following object types:

- Data domain
- Masking rule
- Generation rule
- Policy
- Project
- Plan

The ilmcmd delete command uses the following syntax:

```
ilmcmd
<-Delete | -d>
<-ObjectType | -ot> object_type
<-Project> project_name
<-Name | -n> object_name
[-Help | -h]
```

The following table describes ilmcmd delete options and arguments:

Option	Argument	Description
-ObjectType -ot	object_type	Required to delete an object by name. The type of object to delete. For example, you can enter "Project."
-Project	project_name	Required to delete an object by name. The name of the project that contains the object to delete.
-Name -n	object_name	Required to delete an object by name. The name of the object that you want to delete.
-Help -h	n/a	Optional. Displays help for the command.

Delete Examples

Deleting a Masking Rule

The following sample command deletes the `Credit_Card_Mask` rule on UNIX:

```
./ilmcmd -d -ot MaskingRule -n Credit_Card_Mask
```

Deleting a Policy

The following sample command deletes the `Personal` policy on UNIX:

```
./ilmcmd -d -ot Policy -n Personal
```

Deleting a Generation Rule

The following sample command deletes the `SIN_RandomNumeric` rule on Windows:

```
ilmcmd -d -ot GenerationRule -n SIN_RandomNumeric
```

Deleting a Project

The following sample command deletes the `Customer_DataGeneration` project on Windows:

```
ilmcmd -d -ot Project -n Customer_DataGeneration
```

Export

Exports objects from the Test Data Manager to an XML file.

When you export an object from the TDM repository, you specify the object type and the location of the object in the repository. You can export the following object types:

- Masking rule
- Generation rule
- Policy
- Data domain
- Entity
- Group
- Template
- Profile
- Project
- Data source
- Source definition
- Connections
- Plan

The ilmcmd export command uses the following syntax:

```
ilmcmd
<-Export | -e>
<-ObjectType | -ot> object_type
<-Name | -n> object_name
<-Project> project_name
<-File | -f> XML_file_location
<-DataSourceName | -dsn> schema_name
[-Help | -h]
```

The following table describes the ilmcmd export options and arguments:

Option	Argument	Description
-ObjectType -ot	object_type	Required. The type of object to export. For example, you can enter "Group" or "RuleAssignment." If the object type contains a space in it, remove the space when you enter the argument.
-Name -n	object_name	Required. The name of the object you want to export.
-Project -p	project_name	Required to export a plan, entity, group, or template. The name of the project that contains the plan or port assignment to export.

Option	Argument	Description
-File -f	XML_file_location	Required. The path and file name of the XML file to which you export the object.
-DataSourceName -dsn	schema_name	Required to export a port assignment. The name of the schema that contains the port assignment to export.
-Help -h	n/a	Optional. Displays help for the command.

Export Examples

Exporting a Policy

The following sample command exports the `Policy_Customer` policy to the `Policy_Customer.xml` file:

```
ilmcmd -e -ot Policy -n Policy_Customer -f C:\Informatica\ILMServer\CLI
\Policy_Customer.xml
```

Exporting a Masking Rule

The following sample command exports the Credit_Card_Mask rule to the `CCR.xml` file:

```
ilmcmd -e -ot MaskingRule -n Credit_Card_Mask -f /home/infa1/Desktop/CCR.xml
```

Exporting a Generation Rule

The following sample command exports the SIN_RandomNumeric rule to the `SINRN.xml` file:

```
ilmcmd -e -ot GenerationRule -n SIN_RandomNumeric -f E:\rules\SINRN.xml
```

Exporting a Project

The following sample command exports the CustRecord_Data project to the `CustRecord.xml` file:

```
ilmcmd -e -ot Project -n CustRecord_Data -f E:\project\CustRecord.xml
```

Import

Imports objects from an XML file to the Test Data Manager.

You can import the following object types:

- Masking rule
- Generation rule
- Policy
- Data domain
- Entity
- Group
- Template
- Profile
- Project

- Data source
- Source definition
- Plan

The ilmcmd import command uses the following syntax:

```
ilmcmd
<-Import | -i>
<-File | -f> XML_file_location
[-Help | -h]
```

The following table describes the ilmcmd import options and arguments:

Option	Argument	Description
-File -f	XML_file_location	Required. The path and file name of the import file.
-Help -h	n/a	Optional. Displays help for the command.

Import Examples

Importing a Plan

The following sample command imports the plan listed in the `Plan_Customer.xml` file to the repository:

```
ilmcmd -i -f C:\Informatica\ILMServer\CLI\Plan_Customer.xml -un Administrator -pd
Administrator -hn ilmserver -port 6002
```

Importing a Source Definition

The following sample command imports the source definition listed in the `SrcDef_CustomerDB.xml` file to the repository:

```
ilmcmd -i -f C:\Informatica\ILMServer\CLI\SrcDef_CustomerDB.xml -un Administrator -pd
Administrator -hn ilmserver -port 6002
```

Importing a Generation Rule

The following sample command imports the generation rule listed in the `SIN_RandomNumeric.xml` file to the repository:

```
ilmcmd -i -f E:\Rules\SIN_RandomNumeric.xml
```

Search

Searches for source definitions, masking rules, policies, and plans.

You can search for the following object types in the repository:

- Masking rule
- Policy
- Data domain
- Entity

- Group
- Template
- Profile
- Project
- Data source
- Source definition
- Connection
- Plan

You cannot use ilmcmd to search data generation objects.

The ilmcmd search command uses the following syntax:

```
ilmcmd
<-Search | -s>
<-ObjectType | -ot> object_type
<-Project | -p> project_name
<-DataSourceName | -dsn> schema_name
<-NamePattern | -np> name_pattern
[-Help | -h]
```

The following table describes the ilmcmd search options and arguments:

Option	Argument	Description
-ObjectType -ot	object_type	Required. The type of object to export. For example, you can enter "Group" or "Project." If the object type contains a space in it, remove the space when you enter the argument.
-Project -p	folder_name	Required to search for a source definition, plan, entity, group, or template. The name of the project that contains the object to search for.
-DataSourceName -dsn	schema_name	Required to search for a source definition. The name of the schema that contains the source definition.
-NamePattern -np	name_pattern	Required. The name pattern. Use the asterisk (*) character as a wild card. For example, the name pattern `Rule_C*` returns the following rules: - `Rule_Customer` - `Rule_CustID`
-Help -h	n/a	Optional. Displays help for the command.

Search Examples

Searching for a Masking Rule

The following sample command searches for rules that match the name pattern `Rule*`:

```
ilmcmd -s -ot Rule -np Rule.*
```

The sample command returns the following output:

```
Rule_Customer
Rule_Ticket
```

Searching for a Policy

The following sample command searches for policies that match the name pattern `CUST*`:

```
ilmcmd -s -ot Policy -np CUST.*
```

The sample command returns the following output:

```
CUSTOMER
CUST_COUPONS
```

Workflow

Generates a workflow, displays the status of a workflow run, or runs a workflow.

The ilmcmd workflow command uses the following syntax:

```
ilmcmd
<-WorkFlow | -wf>
<<-Generate | -g> |  <-Execute | -ex>  | <-GetStatus | -gs>>
<-Project> folder_name
<-PlanName | -pn> plan_name
<-ParamFile | -pf paramfile
<-WorkFlowName | -wfn> Workflow_name
<-IntegrationService | -is> integration_service_name
[-Help | -h]
```

The following table describes the ilmcmd workflow options and arguments:

Option	Argument	Description
-Generate -g	n/a	Generates a workflow.
-Execute -ex	n/a	Runs the workflow.
-GetStatus -gs	n/a	Gets the status of a workflow.
-GetWorkFlows -gwf	n/a	Gets workflows for a plan.
-Project	Project	Required. The name of the folder that contains the plan.
-PlanName -pn	PlanName	Required. The plan name.
-paramfile -pf	ParamFile	Required if the plan contains parameters and you want to use parameter values from a specific parameter file. If you do not enter this option, the workflow uses the parameter file specified in the plan.
-WorkFlowName -wfn	WorkflowName	Required to generate a workflow. The location of the workflow properties file.

Option	Argument	Description
-IntegrationService -is	IntegrationService	Required to run a workflow. The name of the PowerCenter Integration Service. Required when you run the workflow.
-JobId	JobId	Required. The job ID of the workflow run.
-Help -h	n/a	Optional. Displays help for the command.

Workflow Examples

Generating a Workflow

The following sample command generates a workflow for the `Plan_NameMasking` plan:

```
ilmcmd -wf -g -Project Infa_Project -pn Plan_NameMasking
```

Monitoring a Workflow

The following sample command monitors the status of a workflow for a plan:

```
ilmcmd -wf -gs -JobId 360
```

Running a Workflow

The following sample command runs a workflow for the `Plan_NameMasking` plan:

```
ilmcmd -wf -ex -Project TDM -pn Plan_NameMasking -is PCInteg
```

Reset

You can reset a data set version to a target and override the source properties.

The reset command uses the following syntax:

```
ilmcmd
[options]
<-Reset | -r>
<-DatasetVersionName | -dvn> Data set version name
<-ResetFile | -rf> File location
<-ResetIntegrationService | -ris> Integration Service
[-Verbose | -v]
[-Help | -h]
```

The following table describes the reset command options and arguments:

Option	Argument	Description
-Reset -r	n/a	Required. The reset job that you create to reset a data set version.
-DatasetVersionName -dvn	Data set version name	Required. The name of the data set version that you want to reset.

Option	Argument	Description
-ResetFile -rf	File location	Required. The location of the file that contains the reset objects.
-ResetIntegrationService -ris	Integration Service	Required. The name of the Integration Service that runs the reset workflow operation.
-Verbose -v	n/a	Optional. Verbose output. Displays more than the minimum information about the object.
-Help -h	n/a	Optional. Displays or prints the help options.

Reset Example

The following sample command resets a data set version:

```
ilmcmd -r -dvn "RESET Version1" -rf "C:\Documents\ResetObject.json" -ris PIS
```

Sample Reset File Properties

Use a sample reset file to configure the reset properties.

After you install TDM, you can find the `ResetObject.xml` file or the `ResetObject.json` file in the following location:

```
<Installation directory>/TDM/utilities/ilmcli/conf/
```

The following table describes the reset file properties and the description:

Property	Description
SOURCE_CONN_NAME	The name of the source connection.
TRUNCATE_TRGT_TABLES	The target tables that TDM truncates if you want to completely replace the target connection with data from the data set version. TDM deletes all data in the target, and copies the data set version to the target. TDM disables constraints and indexes when it performs this operation.
AUTOMATIC_DISABLE_CONSTRAINT	Disable the constraints automatically during the reset operation.
MANUAL_DISABLE_CONSTRAINT	Disable the constraints manually during the reset operation.
DISABLE_INDEXES	Disable indexes during the reset operation.
UPSERT_IN_TRANSACTIONAL_TABLES	Performs an upsert on transactional tables instead of an insert. If the parent records of records in the data set version have changed to point to a different parent in the target, the insert might fail. The reset operation fails if the insert fails.
ENABLE_RECOVERY	Enables the reset recovery options.
BACKUP_RECOVERY_PREFIX	A prefix for the backup schema. TDM creates a backup schema in the format prefix_tablename.

Property	Description
BACKUP_RECOVERY_USER_NAME	The name of the schema where you want to store the backup tables.
STOP_ON_ERROR	Indicates how many non fatal errors the PowerCenter Integration Service encounters before it stops the session. If you enter zero, the session does not stop for nonfatal errors. Default is zero.
ENABLE_HIGH_PRECISION	Processes the decimal data type to a precision of 28. Default is No.
MAX_TABLES_PER_MAPPING	Sets the number of tables to process in a workflow. Default is 10.
MAX_PARALLEL_SESSIONS	The maximum number of sessions that can run at the same time. Applicable if you disable indexes and constraints in the target.
LOCALE_CODE	Sets the locale code for data reset operations.
EMAIL_ADDRESS	An email address to send notifications to.
EMAIL_SUBJECT	The subject of the email that you enter.
EMAIL_TEXT	The content of the email that you enter.
TARGET_LOAD_TYPE	The target load type. You can choose Normal or Bulk. If you select Normal, the PowerCenter Integration Service loads targets normally. Choose Normal mode if the mapping contains an Update Strategy transformation. Choose Bulk to improve session performance. Bulk load limits the ability to recover because no database logging occurs. Default is Normal.
ENABLE_SPECIAL_CHARACTERS_IN_METADATA	Enables the support for special characters in metadata.
TARGET_CONN_TYPE	The type of the target connection. You can select the relational connection types such as Oracle, Microsoft SQL Server, DB2, and Sybase.
TARGET_CONN_NAME	The name of the target connection.
TARGET_PREFIX_NAME	The name prefix of the target table. Specify the target schema name if the target is in a different schema.

ListPlans

Lists all the plans within a data set version.

The ListPlans command uses the following syntax:

```
ilmcmd
<-ListPlans | -lp>
<-DatasetVersionName | -dvn> Dataset_Version_Name
[-Verbose | -v]
[-Help | -h]
```

The following table describes the ListPlans command options and arguments:

Option	Argument	Description
-DatasetVersionName -dvn	Dataset_Version_Name	Required. The name of the data set version from which you want to list the plans.
-Verbose -v	n/a	Optional. Verbose output. Displays more than the minimum information about the object.
-Help -h	n/a	Optional. Displays or prints the help options.

ListPlans Example

The following sample command lists all the plans in a data set version:

```
ilmcmd -lp -dvn "Customer_dataset Version1"
```

TDWPlanGenerate

Generates a workflow to create a subset of a data set.

The TDWPlanGenerate command uses the following syntax:

```
ilmcmd
<-TDWPlanGenerate | -gp>
<-DatasetVersionName | -dvn> Dataset_Version_Name
<-PlanName | -pn> plan_name
[-Verbose | -v]
[-Help | -h]
```

The following table describes the TDWPlanGenerate options and arguments:

Option	Argument	Description
DatasetVersionName -dvn	Dataset_version_Name	Required. The name of the data set from which you create the subset.
-PlanName -pn	plan_name	Required. The name of the plan.
-Verbose -v	n/a	Optional. Verbose output. Displays more than the minimum information about the object.
-Help -h	n/a	Optional. Displays or prints the help options.

TDWPlanGenerate Example

The following sample command generates a workflow to create a subset of a data set:

```
ilmcmd -gp -dvn "Country_dataset Version1" -pn region_dataset
```

TDWPlanExecute

Runs a plan to create a subset of a data set.

The TDWPlanExecute command uses the following syntax:

```
ilmcmd
<-TDWPlanExecute | -ex>
<-DatasetVersionName | -dvn> Dataset_Version_Name
<-PlanName | -pn> plan_name
<-IntegrationService | -is> PowerCenter_Integration_Service_name
[-Verbose | -v]
[-Help | -h]
```

The following table describes the TDWPlanExecute options and arguments:

Option	Argument	Description
DatasetVersionName -dvn	Dataset_version_Name	Required. The name of the data set from which you create the subset.
-PlanName -pn	plan_name	Required. The name of the plan.
IntegrationService -is	PowerCenter_Integration_Service_Name	Required. The name of the PowerCenter Integration Service that runs the workflow.
-Verbose -v	n/a	Optional. Verbose output. Displays more than the minimum information about the object.
-Help -h	n/a	Optional. Displays or prints the help options.

TDWPlanExecute Example

The following sample command runs a workflow to create a subset of a data set:

```
ilmcmd -ex -dvn "Country_dataset Version1" -pn region_dataset -is PCIS
```

TDWPlanGenExe

Generates and runs a data set subset plan from a data set version.

The TDWPlanGenExe command uses the following syntax:

```
ilmcmd
<-TDWPlanGenExe | -ge>
<-DatasetVersionName | -dvn> Dataset_Version_Name
<-PlanName | -pn> plan_name
<-IntegrationService | -is> PowerCenter_Integration_Service_Name
[-Verbose | -v]
[-Help | -h]
```

The following table describes the TDWPlanGenExe command options and arguments:

Option	Argument	Description
-DatasetVersionName -dvn	Dataset_Version_Name	Required. The name of the data set version that you want to create a subset of.
-PlanName -pn	plan_name	Required. The name of the data set subset plan.
-IntegrationService -is	PowerCenter_Integration_Service_Name	Required. The name of the PowerCenter Integration Service that runs the workflow.
-Verbose -v	n/a	Optional. Verbose output. Displays more than the minimum information about the object.
-Help -h	n/a	Optional. Displays or prints the help options.

TDWPlanGenExe Example

The following sample command creates a data set subset from a data set version:

```
ilmcmd –ge –dvn “Country_dataset Version1” –pn region_dataset –is PCIS
```

CHAPTER 17

tdwcmd

This chapter includes the following topics:

- tdwcmd Overview, 326
- Running tdwcmd, 326
- Entering Options and Arguments, 327
- Syntax Notation, 327
- List, 328

tdwcmd Overview

tdwcmd is a command line program that you can use to perform test data warehouse tasks.

Use the tdwcmd command to list the test data warehouse repository, test data warehouse, connection, and data set details.

You must have the required privileges to perform these tasks. For tasks related to projects, you must also have the required permission on the project.

Running tdwcmd

Invoke tdwcmd from the command line. You can issue commands directly or from a script, batch file, or other program.

On Windows, tdwcmd is a batch file with a .bat extension. On UNIX, tdwcmd is a shell script with no extension.

1. At the command line, switch to the directory where the tdwcmd executable file is located.

   By default, tdwcmd is installed in the following directory:

   ```
   <TDM installation directory>\utilities\tdwcli\bin
   ```

2. Enter tdwcmd followed by the command name and the required options and arguments.

   For example:

   ```
   tdwcmd -command_name [-option1] argument_1 [-option2] argument_2...
   ```

To view the command line syntax, enter the following command:

```
tdwcmd -h
```

Entering Options and Arguments

The tdwcmd command line program uses a set of options and arguments.

Use the following rules when you enter the tdwcmd command options and arguments:

- To enter options, type a hyphen followed by the program syntax for the command. Use two consecutive hyphens for the long name of the option, and use a single hyphen for the short name of the option.
- You can enter options in any order.
- If an argument contains spaces, enclose the argument in double quotes.
- The first word after the option is the argument.
- Most options require arguments. You must separate options from arguments with a single space.
- Commands, options, and object names are case sensitive.

Syntax Notation

To use the tdwcmd command line program, review the syntax notation.

The following table describes the syntax notation for the tdwcmd command line program:

Convention	Description
--x or -x	An option placed before an argument. The option designates the parameter that you enter. For example, to enter the user name for tdwcmd, type --UserName or -un followed by the user name.
<x>	Required option. If you omit a required option, the command line program returns an error message.
<x | y>	Select between required options. If you omit a required option, the command line program returns an error message. If a pipe symbol (|) separates options, specify one option. If there are no pipe symbols to separate the options, specify all the options.
[x]	Optional option. The command runs whether or not you enter these options.
[x | y]	Select between optional options. For example, you can display help for all tdwcmd commands by using the --Help or -h option. `[--Help|-h]` The command runs whether or not you enter the optional parameter.
< <x | y> | <a | b> **>**	When a set contains subsets, indicate the supersets with bold brackets (**< >**). Separate the subsets with the bold pipe symbol (|).

List

You can search and list test data warehouse repository, connection, test data warehouse, and data set details.

You can search for the following object types in the repository:

- Test data warehouse repository
- Test data warehouse
- Connection
- Data set
- Data set version
- Data set version tables

The list command uses the following syntax:

```
tdwcmd
<--list | -list>
[--Repository | -Repository]
[--TestDataWarehouse | -TestDataWarehouse]
[--Connection | -Connection]
<--Dataset | -Dataset>
[--TestDataWarehouseName | -twn] TestDataWarehouseName
<--DatasetVersion | -DatasetVersion>
[--TestDataWarehouseName | -twn] TestDataWarehouseName
[--DatasetName | -dn] DatasetName
<--DatasetVersionTable | -DatasetVersionTable>
[--TestDataWarehouseName | -twn] TestDataWarehouseName
[--TableDatasetName | -tdn] TableDatasetName
[--DatasetVersionNumber | -dvn] DatasetVersionNumber
<--HostName | -host_name] HostName
[--HTTPPort | -http_port] HTTPPort
[--HTTPSPort | -https_port] HTTPSPort
<--UserName | -un> UserName
<--Password | -pd> Password
<--SecurityDomain | -sdn> SecurityDomain
<--DomainName | -dn> DomainName
<--TDWServiceName | -tsn> TDWServiceName
[--Verbose | -v]
[--Help | -?, -h]
```

The following table describes the List command options and arguments:

Option	Argument	Description
--Repository -Repository	n/a	The test data warehouse repository.
--TestDataWarehouse -TestDataWarehouse	n/a	The relational database that stores the data related to data sets that you store in the test data warehouse.
--Connection -Connection	n/a	The connection to the data warehouse repository.
--Dataset -Dataset	n/a	Required. The data set that you create in the test data warehouse.
--TestDataWarehouseName -twn	TestDataWarehouseName	The name of the test data warehouse.

Option	Argument	Description
--DatasetVersion -DatasetVersion	n/a	Required. The version of the data set.
--DatasetName -dn	DatasetName	The name of the data set.
--DatasetVersionTable -DatasetVersionTable	n/a	Required. The version of the data set table.
--TableDatasetName -tdn	TableDatasetName	The name of the data set.
--DatasetVersionNumber -dvn	DatasetVersionNumber	The version number of the data set.
--HostName -host_name	HostName	Required. The name of the host that runs the Test Data Warehouse Service.
--HTTPPort -http_port	HTTPPort	Required if you do not specify an HTTPS port. Port number to connect to the Test Data Warehouse Service.
--HTTPSPort -https_port	HTTPSPort	Required if you do not specify an HTTP port. Port number to connect to the Test Data Warehouse Service.
--UserName -un	UserName	Required. The Informatica domain user name.
--Password -pd	Password	Required. The password for the Informatica domain user.
--SecurityDomain -sdn	SecurityDomain	Required. The Informatica security domain.
--DomainName -dn	DomainName	Required. The name of the Test Data Warehouse domain.
--TDWServiceName -tsn	TDWServiceName	Required. The name of the Test Data Warehouse Service.
--Verbose -v	n/a	Optional. Verbose output. Displays more than the minimum information about the object.
--Help -?, -h	n/a	Optional. Displays or prints the help options.

List Examples

Listing a Test Data Warehouse Repository

The following sample command lists the test data warehouse repository details:

```
tdwcmd --list --Repository --HostName localhost --HTTPPort 6605 -un Administrator -pd
Administrator -sdn Native -dn  Domain_SandboxVMLin_TDM961HF14 --TDWServiceName TDWS
```

Listing a Test Data Warehouse

The following sample command lists the test data warehouse details:

```
tdwcmd --list --TestDataWarehouse --HostName localhost --HTTPPort 6605 -un Administrator
-pd Administrator -sdn Native -dn  Domain_SandboxVMLin_TDM961HF14 --TDWServiceName TDWS
```

Listing a Connection

The following sample command lists the connection details:

```
tdwcmd --list --Connection --HostName localhost --HTTPPort 6605 -un Administrator -pd
Administrator -sdn Native -dn  Domain_SandboxVMLin_TDM961HF14 --TDWServiceName TDWS
```

Listing a Data Set

The following sample command lists the data set details:

```
tdwcmd --list --Dataset -twn OracleWarehouse --HostName localhost --HTTPPort 6605 -un
Administrator -pd Administrator -sdn Native -dn  Domain_SandboxVMLin_TDM961HF14 --
TDWServiceName TDWS
```

Listing a Data Set Version

The following sample command lists the data set versions:

```
tdwcmd --list --DatasetVersion -twn OracleWarehouse --DatasetName Entity --HostName
localhost --HTTPPort 6605 -un Administrator -pd Administrator -sdn Native -dn
Domain_SandboxVMLin_TDM961HF14 --TDWServiceName TDWS
```

Listing a Data Set Version Table

The following sample command lists the data set version tables:

```
tdwcmd --list --DatasetVersionTable -twn OracleWarehouse -tdn Entity -dvn 1 -host_name
localhost -http_port 6605 --UserName Administrator --Password Administrator --
SecurityDomain Native -dn  Domain_SandboxVMLin_TDM961HF14 --TDWServiceName TDWS
```

CHAPTER 18

tdwquery

This chapter includes the following topics:

- tdwquery Overview, 331
- Configuring tdwquery, 331
- Running tdwquery, 332
- Select Clause, 333

tdwquery Overview

tdwquery is a command line program that you can use to analyze data in a data set that you store in the test data warehouse.

You can run queries to view data in a data set. You can run queries on tables in a single data set at a time. You can use multiple joins, functions, and clauses to filter and search for data when you run queries.

Before you run a query, you must configure the properties file.

Configuring tdwquery

You must configure the tdwquery properties file before you can run a query.

1. Browse to the following location and find the properties file `userConfig.ilm`:

   `<Informatica installation directory>/TDM/utilities/tdwquery/conf/`

2. Use a text editor to open the file and enter the values of the properties in the file.

You must configure the following properties:

Property	Description
TDW_SERVICE_URL	The URL of the Test Data Warehouse Service that you want to connect to. For example, *http://testhost:7705/tdw*.
USERNAME	Name of the test data warehouse user that runs the query. The user must have the Test Data Warehouse Service View Data in Data Set privilege and at least read permission on the data set.
PASSWORD	Password of the test data warehouse user.
DOMAINNAME	Name of the domain to which the test data warehouse user belongs.
NAMESPACE	Namespace of the test data warehouse user.
TDWSERVICENAME	Name of the Test Data Warehouse Service.
TESTDATAWAREHOUSENAME	Name of the test data warehouse where you stored the data set.
DATASETVERSIONNAME	Name of the data set version that you want to run a query on.
PAGENUMBER	Required if you enter a value for PAGESIZE. The number of the page that you want to view.
PAGESIZE	Required if you enter a value for PAGENUMBER. The number of records that you want to appear on each page.

Running tdwquery

Run tdwquery from the command line. On Windows, tdwquery is a batch file with a .bat extension. On UNIX, tdwquery is a shell script with no extension.

You must configure the tdwquery properties file before you run a query.

1. At the command line, switch to the directory where the tdwquery executable is located.

   By default, tdwquery is installed in the following directory:
   `<Informatica installation directory>/TDM/utilities/tdwquery`.

2. To run a query:

   - On Windows, enter tdwquery followed by the clause and the required syntax.
     For example:

     ```
     tdwquery "SELECT * FROM td_empdetails"
     ```

     You can also prefix the schema name to the clause. For example:

     ```
     tdwquery "SELECT * FROM AA.td_empdetails"
     ```

   - On UNIX, enter ./ tdwquery followed by the clause and the required syntax.
     For example:

     ```
     ./tdwquery "SELECT * FROM td_empdetails"
     ```

You can also prefix the schema name to the clause. For example:

```
./tdwquery "SELECT * FROM AA.td_empdetails"
```

Select Clause

You can use the Select and Select Distinct clauses to run queries to analyze data in a table.

For example, the following query returns all data in table1:

```
SELECT * from table 1
```

The following query returns distinct data values from the columns col 1 and col 2 in table 2.

```
SELECT DISTINCT col 1, col 2 from table 2
```

You can use joins, aggregate functions, the Where clause with options, and other clauses with the Select and Select Distinct clauses.

You can run queries with table names and column names that contain spaces or dashes. You must enter the table name or column name within '\"'. Use the following syntax:

```
SELECT \"<column-name>\" from \"<table name>\"
```

Note:

- You cannot use other special characters in a query.
- You can use only English characters, the underscore ('_'), and numeric characters in a query.

Alias

You can use aliases when you run a query with the Select and Select Distinct clauses.

For example:

```
SELECT <alias_name>.<column_name> FROM <table_name> <alias_name>

SELECT <column_name> as <alias_name> from <table_name>

SELECT e.dept_id FROM Employee e

SELECT emp_name as name from TDWEMP

SELECT e.dept_id, e.name FROM Employee e WHERE e.name like 'J%'
```

Joins

You can use joins when you run a query with the Select and Select Distinct clauses.

You can use the following types of joins in a query:

Inner Join

Use the following syntax to perform an inner join:

```
SELECT <columns>
FROM <table 1>
INNER JOIN <table 2>
ON <table 1>.<column_name> = <table 2>.<column_name>
```

For example:

```
SELECT suppliers.supplier_id, suppliers.supplier_name, orders.order_date
FROM suppliers,
INNER JOIN orders
ON suppliers.supplier_id = orders.supplier_id
```

Left Outer Join

Use the following syntax to perform a left outer join:

```
SELECT <columns>
FROM <table 1>
LEFT OUTER JOIN <table 2>
ON <table 1>.<column_name> = <table 2>.<column_name>
```

For example:

```
SELECT suppliers.supplier_id, suppliers.supplier_name, orders.order_date
FROM suppliers
LEFT OUTER JOIN orders
ON suppliers.supplier_id = orders.supplier_id
```

Right Outer Join

Use the following syntax to perform a right outer join:

```
SELECT <columns>
FROM <table 1>
RIGHT OUTER JOIN <table 2>
ON <table 1>.<column_name> = <table 2>.<column_name>
```

For example:

```
SELECT orders.order_id, orders.order_date, suppliers.supplier_name
FROM suppliers
RIGHT OUTER JOIN orders
ON suppliers.supplier_id = orders.supplier_id
```

Full Outer Join

Use the following syntax to perform a full outer join:

```
SELECT <columns>
FROM <table 1>
FULL OUTER JOIN <table 2>
ON <table 1>.<column_name> = <table 2>.<column_name>
```

For example:

```
SELECT suppliers.supplier_id, suppliers.supplier_name, orders.order_date
FROM suppliers
FULL OUTER JOIN orders
ON suppliers.supplier_id = orders.supplier_id
```

Aggregate Functions

You can use aggregate functions when you run a query with the Select and Select Distinct clauses.

You can use the following types of aggregate functions in a query:

AVG

Use the following syntax:

```
SELECT AVG(<column_name>) FROM <table_name>
```

For example:

```
SELECT AVG(SAL) FROM TDWEMP
```

COUNT

Use the following syntax:

```
SELECT COUNT(<column_name>) FROM <table_name>
```

For example:

```
SELECT COUNT(*) FROM TDWEMP
```

MAX

Use the following syntax:

```
SELECT MAX(<column_name>) FROM <table_name>
```

For example:

```
SELECT MAX(SAL) FROM TDWEMP
```

MIN

Use the following syntax:

```
SELECT MIN(<column_name>) FROM <table_name>
```

For example:

```
SELECT MIN(SAL) FROM TDWEMP
```

SUM

Use the following syntax:

```
SELECT SUM(<column_name> FROM <table_name>
```

For example:

```
SELECT SUM(SAL) FROM TDWEMP
```

Where Clause

You can use the Where clause to specify filter criteria in a query with the Select and Select Distinct clauses.

You can run queries that use parentheses with the Where clause. For example:

```
SELECT * from Employee where NAME ='Mary' OR (NAME='Jessica' AND DEPT_ID=1)
```

You can use the Where clause with the following conditions in a query:

Single condition

Use the following syntax:

```
SELECT * FROM <table_name> WHERE <column_name> = <column_value>
```

For example:

```
SELECT * FROM customers WHERE state = 'California'
```

And

Use the following syntax:

```
SELECT * FROM <table_name> WHERE <column_name> = <column_value> AND <column_value>
<operator> <column_value>
```

For example:

```
SELECT * FROM customers WHERE state = 'California' AND company_name = 'Informatica'
```

OR

Use the following syntax:

```
SELECT * FROM <table_name> WHERE <column_name> = <column_value> OR <column_name>
<operator> <column_value>
```

For example:

```
SELECT * FROM customers WHERE state = 'California' OR company_name = 'Informatica'
```

Combining OR and AND

Use the following syntax:

```
SELECT * FROM <table_name> WHERE <column_name> = <column_value> OR <column_name>
<operator> <column_value> AND <column_name> <operator> <column_value>
```

For example:

```
SELECT * FROM customers WHERE state = 'California' OR available_credit > 500 and
revenue < 90
```

Joining and where

Use the following syntax:

```
SELECT columns FROM <table 1> LEFT OUTER JOIN <table 2> ON <table 1>.<col 1> =
<table 2>.<col 1> WHERE <table 1>.<col 1> <operator> <column_value>
```

For example:

```
SELECT suppliers.supplier_id, suppliers.supplier_name, orders.order_date FROM
suppliers LEFT OUTER JOIN orders ON suppliers.supplier_id = orders.supplier_id WHERE
supplier.supplier_id > 10
```

Exists

Use the following syntax:

```
SELECT DISTINCT <col 1> FROM <table 2> WHERE EXISTS (SELECT * FROM <table 2> WHERE
<table 1>.<col 1> <operator> <table 1>.<col 1>)
```

For example:

```
SELECT DISTINCT store_type FROM stores WHERE EXISTS (SELECT * FROM cities_stores
WHERE cities_stores.store_type = stores.store_type)
```

In

Use the following syntax:

```
SELECT * FROM <table 1> WHERE <col 1> IN (SELECT <col 1> FROM <table 2>)
```

For example:

```
SELECT * FROM Employee WHERE dept_id IN (SELECT DEPT_ID FROM TDWDEPT)
```

Like

Use the following syntax:

```
SELECT * FROM <table 1> WHERE <col 1> LIKE <value>
```

You can use characters or wildcards '%' and '_'. For example:

```
SELECT * FROM Employee WHERE name LIKE 'J%'
```

Is Null or Is Not Null

Use the following syntax:

```
SELECT * FROM <table 1> WHERE <col 1> IS NULL
```

For example:

```
SELECT * FROM Employee WHERE ssn IS NULL
```

Between

Use the following syntax:

```
SELECT * FROM <table 1> WHERE <col 1> BETWEEN <value> AND <value>
```

For example:

```
SELECT * FROM Employee WHERE delt_id BETWEEN 2 AND 3
```

Not

Use the following syntax:

```
SELECT * FROM <table 1> WHERE <col 1> NOT <CONDITION> <value>
```

For example:

```
SELECT DISTINCT * from STATE s WHERE NOT EXISTS (SELECT ct.state_id from city ct
where ct.state_id = s.state_id)
```

```
SELECT * FROM Employee WHERE dept_id NOT IN (SELECT DEPT_ID FROM TDWDEPT)
```

```
SELECT * FROM Employee WHERE name NOT LIKE 'J%'
```

```
SELECT * FROM Employee WHERE delt_id NOT BETWEEN 2 AND 3
```

Additional Clauses

You can use the following clauses when you run a query with the Select clause:

Group by

Use the following syntax:

```
SELECT <col 1> FROM <table 1> GROUP BY <col 1>
```

For example:

```
SELECT name FROM Employee GROUP BY NAME
```

Having

Use the following syntax:

```
SELECT <col 1> FROM <table 1> GROUP BY <col 1> HAVING <col 1> LIKE <value>
```

For example:

```
SELECT name FROM Employee GROUP BY NAME HAVING name like 'J%'
```

Order by

Use the following syntax. You can use the optional modifiers ASC and DESC:

```
SELECT <col 1> FROM <table 1> ORDER BY <col 1> ASC
```

For example:

```
SELECT dept_id, name FROM Employee ORDER BY dept_id ASC
```

APPENDIX A

Data Type Reference

This appendix includes the following topics:

- Data Type Reference Overview, 338
- Oracle, 339
- Microsoft SQL Server, 343
- DB2 for Linux, UNIX, and Windows, 347
- Sybase ASE, 349
- Hadoop HDFS, 352
- MySQL, 353
- Flat File, 354
- VSAM Flat, 355
- DB2 for z/OS, 356
- DB2 for IOS, 358
- IMS Flat, 360
- Sybase IQ, 361
- Netezza, 363
- Teradata, 365
- Cassandra, 367
- MongoDB, 368

Data Type Reference Overview

You can pass different data types from sources to targets and perform TDM operations on the data types.

In TDM, you can perform data masking, data subset, and data discovery operations on the source databases for supported data types. You cannot perform the operations on data types that TDM does not support. For some data types, you might need to configure the precision and scale to a specific range of values.

Oracle

The following table describes the operations that you can perform on data types in Oracle database:

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Binary_Double	Yes	Yes	Yes	Yes	Yes	
Binary_Float	Yes	Yes	Yes	Yes	Yes	
Blob	Yes	Yes	Yes	Yes	No	You can perform nullification masking.
Char	Yes	Yes	Yes	Yes	Yes	
Clob	Yes	Yes	Yes	Yes	No	
Date	Yes	Yes	Yes	Yes	Yes	You cannot run data domain profiles.
Dburitype	No	No	No	Yes	No	Use dictionary generation rule to generate data.
Decimal(P,S)	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Float	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles. To generate test data, you enter a precision value less than 15.
Httpuritype	No	No	No	Yes	No	Use dictionary generation rule to generate data.
Integer	Yes	Yes	Yes	Yes	Yes	

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Interval day to second	Yes	Yes	No	No	No	You can enter the following criteria: Is Null, Is Not Null, Like, Not Like, Is Empty, and Is Not Empty. For the Interval day(9) to second(9) data type, you must set the precision greater than 18.
Interval year to month	Yes	Yes	No	Yes	No	You can enter the following entity criteria: Is Null, Is Not Null, Like, Not Like, Is Empty, and Is Not Empty. For the Interval year(9) to month data type, you must set the precision greater than 11. Use dictionary generation rule to generate data.
Long	Yes	Yes	Yes	Yes	No	In a sequence generation rule, you cannot choose to start the sequence from the last output value.
Longraw	Yes	Yes	Yes	Yes	No	You can perform nullification masking. Use dictionary generation rule to generate data.

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Longvarchar	Yes	Yes	Yes	Yes	No	In a sequence generation rule, you cannot choose to start the sequence from the last output value.
Nchar	Yes	Yes	Yes	Yes	Yes	
Nclob	Yes	Yes	Yes	Yes	No	Use string generation rule to generate data.
Number	Yes	Yes	Yes	Yes	Yes	
Nvarchar	Yes	Yes	Yes	Yes	Yes	
Nvarchar2(N)	Yes	Yes	Yes	Yes	Yes	
Raw	Yes	Yes	Yes	Yes	No	You can perform nullification masking. Use dictionary generation rule to generate data.
Real	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Rowid	Yes	Yes	No	No	No	
Smallint	Yes	Yes	Yes	Yes	Yes	
Timestamp	Yes	Yes	Yes	Yes	Yes	You cannot run data domain profiles.
Timestamp with time zone	Yes	Yes	No	No	No	You can enter the following entity criteria: Is Null, Is Not Null, Like, Not Like, Is Empty, and Is Not Empty.
Urowid	Yes	Yes	No	No	No	
Varchar	Yes	Yes	Yes	Yes	Yes	
Varchar2(N)	Yes	Yes	Yes	Yes	Yes	

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
XML	Yes	Yes	No	Yes	No	You can enter the following entity criteria: Is Null, Is Not Null, Is Empty, and Is Not Empty. If you assign a dictionary generation rule that contains specified invalid values for the XML data type in an Oracle database, the data generation task fails.
Xdburitype	No	No	No	Yes	No	Use dictionary generation rule to generate data.

You cannot perform data masking, data subset, profiling, and data generation operations on the following Oracle data types:

- Anydata
- Anydataset
- Anytype
- Bfile
- Ordaudio
- Orddicom
- Orddoc
- Ordimage
- Ordimagesignature
- Ordvideo
- Sdo_Geometry
- Sdo_Georaster
- Sdo_Topo_Geometry
- SI_Averagecolor
- SI_Color
- SI_Colorhistogram
- SI_Featurelist
- SI_Positionalcolor
- SI_Stillimage
- SI_Texture

- Timestamp with local time zone
- Uritype

Microsoft SQL Server

The following table describes the operations that you can perform on data types in Microsoft SQL Server database:

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Bigint	Yes	Yes	Yes	Yes	Yes	
Binary	Yes	Yes	Yes	Yes	No	Use dictionary generation rule to generate test data. Use nullification data masking rule to mask the data.
Bit	Yes	Yes	Yes	Yes	Yes	If you enter data patterns, you must provide T and F values instead of 1 and 0.
Char	Yes	Yes	Yes	Yes	Yes	
Date	Yes	Yes	Yes	Yes	Yes	You can primary profiles.
Datetime	Yes	Yes	No	Yes	Yes	You cannot run data domain profiles.
Datetime2	Yes	Yes	Yes	Yes	No	
Datetimeoffset	Yes	Yes	No	Yes	No	
Decimal	Yes	Yes	Yes	Yes	Yes	You can enter a precision value up to 28.

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Float	Yes	Yes	Yes	Yes	Yes	To perform a data masking operation, you must set the precision to less than 28. When you run a test data generation operation, TDM truncates data if the precision value is greater than 17. You cannot run entity profiles.
Image	Yes	Yes	Yes	Yes	No	
Int	Yes	Yes	Yes	Yes	Yes	
Money	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Nchar	Yes	Yes	Yes	Yes	Yes	
Ntext	Yes	Yes	Yes	Yes	No	In a sequence generation rule, you cannot choose to start the sequence from the last output value.
Numeric	Yes	Yes	Yes	Yes	Yes	You can enter a precision value up to 28.
Nvarchar	Yes	Yes	Yes	Yes	Yes	
Real	Yes	Yes	Yes	Yes	Yes	When you run a test data generation operation, TDM truncates data if the precision is greater than 15. You cannot run entity profiles.
Smalldatetime	Yes	Yes	No	Yes	Yes	You cannot run data domain profiles.

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Smallint	Yes	Yes	Yes	Yes	Yes	
Smallmoney	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Sql_variant	Yes	Yes	No	Yes	No	You can enter the following entity criteria: Is Null, Is Not Null, Is Empty, and Is Not Empty. In a sequence generation rule, you cannot choose to start the sequence from the last output value. You must increase the precision value after you import metadata.
Text	Yes	Yes	Yes	Yes	No	In a sequence generation rule, you cannot choose to start the sequence from the last output value.
Time	Yes	Yes	Yes	Yes	Yes	When you enter values in milliseconds, TDM truncates the data. You can run primary key profiles.
Timestamp	n/a	n/a	n/a	Yes	No	
Tinyint	Yes	Yes	Yes	Yes	Yes	

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Uniqueidentifier	Yes	Yes	No	Yes	Yes	You can enter the following entity criteria: Is Null, Is Not Null, Like and Not Like. You must not assign a data generation rule. You cannot run entity and data domain profiles.
Varbinary	Yes	Yes	Yes	Yes	No	Use dictionary generation rule to generate test data. Use nullification data masking rule to mask the data.
Varchar	Yes	Yes	Yes	Yes	Yes	
XML	Yes	Yes	No	Yes	No	You can enter the following entity criteria: Is Null and Is Not Null. Use dictionary generation rule to generate data.

You cannot perform data masking, data subset, profiling, and data generation operations on the following Microsoft SQL Server data types:

- Hierarchyid
- Geography
- Geometry

DB2 for Linux, UNIX, and Windows

The following table describes the operations that you can perform on data types in DB2 database for Linux, UNIX, and Windows operating systems:

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Data Discovery	Comments
Bigint	Yes	Yes	Yes	Yes	Yes	
Blob (N)	Yes	Yes	Yes	Yes	No	You can perform nullification masking. Use dictionary generation rule to generate data.
Char for bit data	Yes	Yes	Yes	Yes	No	
Char varying (Length)	Yes	Yes	Yes	Yes	Yes	To run a profile, you must keep the field length less than or equal to 255.
Character (N)	Yes	Yes	Yes	Yes	Yes	
Clob (N)	Yes	Yes	Yes	Yes	No	You can perform nullification masking.
Date	Yes	Yes	Yes	Yes	Yes	
Dbclob (N)	Yes	Yes	Yes	Yes	No	You can perform nullification masking.
Dec	Yes	Yes	Yes	Yes	Yes	
Decfloat (16)	Yes	Yes	Yes	Yes	No	
Decfloat (34)	Yes	Yes	Yes	Yes	No	
Decimal	Yes	Yes	Yes	Yes	Yes	
Double	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Double Precision	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Data Discovery	Comments
Float	Yes	Yes	Yes	Yes	Yes	You can enter a precision value up to 28. You cannot run entity profiles.
Graphic (N)	Yes	Yes	Yes	Yes	Yes	
Integer	Yes	Yes	Yes	Yes	Yes	
Long Varchar	Yes	Yes	Yes	Yes	No	In a sequence generation rule, you cannot choose to start the sequence from the last output value.
Long Vargraphic	Yes	Yes	Yes	Yes	No	In a sequence generation rule, you cannot choose to start the sequence from the last output value.
Num	Yes	Yes	Yes	Yes	Yes	
Numeric	Yes	Yes	Yes	Yes	Yes	
Real	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Smallint	Yes	Yes	Yes	Yes	Yes	
Time	Yes	Yes	Yes	Yes	Yes	
Timestamp	Yes	Yes	Yes	Yes	Yes	
Varchar for Bit Data	Yes	Yes	Yes	Yes	No	
Varchar (N)	Yes	Yes	Yes	Yes	Yes	To run a profile, you must keep the field length less than or equal to 255.

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Data Discovery	Comments
Vargraphic (N)	Yes	Yes	Yes	Yes	Yes	
XML	Yes	Yes	No	Yes	No	You can enter the following entity criteria: Is Null and Is Not Null. Use dictionary generation rule to generate data.

Sybase ASE

The following table describes the operations that you can perform on data types in Sybase ASE database:

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Bigdatetime	Yes	Yes	Yes	Yes	Yes	
Bigint	Yes	Yes	Yes	Yes	Yes	
Bigtime	Yes	Yes	Yes	Yes	Yes	
Bit	Yes	Yes	Yes	Yes	Yes	You can enter the following entity criteria: Is Null, Is Not Null, Like, and Not Like.
Binary (n)	Yes	Yes	Yes	Yes	No	You can perform nullification masking.
Char (n)	Yes	Yes	Yes	Yes	Yes	You can enter characters up to 2000. To run a profile, you must enter the field length less than or equal to 255.
Date	Yes	Yes	Yes	Yes	Yes	
Datetime	Yes	Yes	Yes	Yes	Yes	

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Decimal (P,S)	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles. You can enter a precision value up to 28.
Double Precision	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles. You can perform data masking for precision values up to 28.
Float (P)	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles. You can perform data masking for precision values up to 28.
Image	Yes	Yes	Yes	Yes	No	Use dictionary generation rule to generate date. You can perform nullification masking.
Int	Yes	Yes	Yes	Yes	Yes	
Longsysnam e	Yes	Yes	Yes	Yes	Yes	
Money	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Nchar (N)	Yes	Yes	Yes	Yes	No	
Numeric (P,S)	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles. You can enter a precision value up to 28.
Nvarchar (N)	Yes	Yes	Yes	Yes	No	
Real	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Smalldateti me	Yes	Yes	Yes	Yes	Yes	

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Smallint	Yes	Yes	Yes	Yes	Yes	
Smallmoney	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Sysname	Yes	Yes	Yes	Yes	Yes	
Text	Yes	Yes	Yes	Yes	No	In a sequence generation rule, you cannot choose to start the sequence from the last output value.
Time	Yes	Yes	Yes	Yes	Yes	
Timestamp	Yes	Yes	Yes	Yes	No	You can perform nullification masking. Timestamp values are system generated in Sybase database. You can restrict the data type from Test Data Manager to generate data.
Tinyint	Yes	Yes	Yes	Yes	Yes	When you assign a data generation rule, ensure that you provide a range between 0 and 255.
Unichar (N)	Yes	Yes	Yes	Yes	Yes	
Unitext	Yes	Yes	Yes	Yes	No	In a sequence generation rule, you cannot choose to start the sequence from the last output value.
Univarchar (N)	Yes	Yes	Yes	Yes	Yes	

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Unsigned bigint	Yes	Yes	Yes	Yes	Yes	You cannot provide negative values.
Unsigned int	Yes	Yes	Yes	Yes	Yes	You cannot provide negative values.
Unsigned smallint	Yes	Yes	Yes	Yes	Yes	You cannot provide negative values.
Unsigned tinyint	Yes	Yes	Yes	Yes	Yes	When you assign a data generation rule, ensure that you provide a range between 0 and 255.
Varbinary (N)	Yes	Yes	Yes	Yes	No	You can perform nullification masking.
Varchar (N)	Yes	Yes	Yes	Yes	Yes	To run a profile, you must enter the field length less than or equal to 255.

Hadoop HDFS

The following table describes the operations that you can perform on data types in a Hadoop HDFS connection:

Data Types	Data Movement	Data Masking	Comments
Datetime	Yes	Yes	You must provide the date time format when you import the sources in the PowerCenter repository.
Number	Yes	Yes	
String	Yes	Yes	

MySQL

The following table describes the operations that you can perform on data types in MySQL database:

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Bigint	Yes	Yes	Yes	Yes	Yes	
Binary	Yes	Yes	Yes	Yes	No	
Bit	Yes	Yes	Yes	Yes	Yes	You cannot run data domain profiles.
Blob	Yes	Yes	Yes	Yes	No	
Char	Yes	Yes	Yes	Yes	Yes	
Date	Yes	Yes	Yes	Yes	Yes	
Datetime	Yes	Yes	Yes	Yes	Yes	
Decimal	Yes	Yes	Yes	Yes	Yes	
Double	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Enum	Yes	Yes	Yes	Yes	Yes	
Float	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Int	Yes	Yes	Yes	Yes	Yes	
Mediumblob	Yes	Yes	Yes	Yes	No	
Mediumint	Yes	Yes	Yes	Yes	Yes	
Mediumtext	Yes	Yes	Yes	Yes	No	TDM allows up to 8 MB row size in a Filter or Sorter transformation. To add a Filter or Sorter transformation in a mapping, you must edit the precision to less than 16,777,215 characters.
Set	Yes	Yes	Yes	Yes	Yes	
Smallint	Yes	Yes	Yes	Yes	Yes	

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Text	Yes	Yes	Yes	Yes	No	
Time	Yes	Yes	Yes	Yes	Yes	
Timestamp	Yes	Yes	Yes	Yes	Yes	
Tinyblob	Yes	Yes	Yes	Yes	No	
Tinyint	Yes	Yes	Yes	Yes	Yes	
Tinytext	Yes	Yes	Yes	Yes	No	
Varbinary	Yes	Yes	Yes	Yes	No	
Varchar	Yes	Yes	Yes	Yes	Yes	
Year	Yes	Yes	Yes	Yes	Yes	TDM generates data for the year range from 1901 to 2155. For any other date range, TDM generates zero in the target database.

You cannot perform a data masking, data subset, profiling, and data generation operation on the following MySQL data types:

- Longblob
- Longtext

Flat File

The following table describes the operations that you can perform on data types in flat files:

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Bigint	Yes	Yes	Yes	Yes	Yes	
Datetime	Yes	Yes	Yes	Yes	Yes	
Double	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Int	Yes	Yes	Yes	Yes	Yes	
Nstring	Yes	Yes	Yes	Yes	Yes	
Number	Yes	Yes	Yes	Yes	Yes	
String	Yes	Yes	Yes	Yes	Yes	

VSAM Flat

The following table describes the operations that you can perform on data types in VSAM:

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Bin	Yes	Yes	No	Yes	No	You can enter the following entity criteria: Is Null and Is Not Null.
Char	Yes	Yes	Yes	Yes	Yes	
Date	Yes	Yes	Yes	Yes	Yes	
Double	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Float	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num16	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num16U	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num32	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num32U	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num64	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num64U	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Num8	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num8U	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Numchar	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Packed	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Spacked	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Time	Yes	Yes	Yes	Yes	Yes	
Timestamp	Yes	Yes	Yes	Yes	Yes	
Upacked	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Uzoned	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Varbin	Yes	Yes	No	Yes	No	You can enter the following entity criteria: Is Null and Is Not Null.
Varchar	Yes	Yes	Yes	Yes	Yes	
Zoned	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.

DB2 for z/OS

The following table describes the operations that you can perform on data types in DB2 for z/OS database:

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Bigint	Yes	Yes	Yes	Yes	No	
Binary	Yes	Yes	Yes	Yes	No	

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Blob	n/a	n/a	n/a	Yes	No	You can generate data through the native client.
Char	Yes	Yes	Yes	Yes	No	
Char for bit data	Yes	Yes	Yes	Yes	No	
Clob	n/a	n/a	n/a	Yes	No	You can generate data through the native client.
Date	Yes	Yes	Yes	Yes	No	
Dbclob	n/a	n/a	n/a	Yes	No	You can generate data through the native client.
Decfloat	n/a	n/a	n/a	Yes	No	You can generate data through the native client.
Decimal	Yes	Yes	Yes	Yes	No	
Double	Yes	Yes	Yes	Yes	No	You can enter a precision value up to 28.
Float	Yes	Yes	Yes	Yes	No	
Graphic	No	No	No	Yes	No	
Integer	Yes	Yes	Yes	Yes	No	
Long Varchar	Yes	Yes	Yes	Yes	No	
Long Varchar for bit data	Yes	Yes	Yes	Yes	No	
Long Vargraphic	No	No	No	Yes	No	
Real	Yes	Yes	Yes	Yes	No	
Smallint	Yes	Yes	Yes	Yes	No	
Time	Yes	Yes	Yes	Yes	No	
Timestamp	Yes	Yes	Yes	Yes	No	

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Varbinary	Yes	Yes	Yes	Yes	No	
Varchar	Yes	Yes	Yes	Yes	No	
Varchar for Bit Data	Yes	Yes	Yes	Yes	No	
Vargraphic	No	No	No	Yes	No	
XML	n/a	n/a	n/a	Yes	No	You can generate data through the native client.

DB2 for IOS

You can perform data masking, data movement, data subset, and data generation operations on the DB2 for IOS data types. You cannot apply constraints on DB2 for IOS databases. To perform data discovery on DB2 for IOS data types, you must import profiles that you create in Informatica Developer.

The following table describes the operations that you can perform on data types in DB2 for IOS database:

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Comments
Bigint	Yes	Yes	Yes	Yes	You must enter a precision value up to 19.
Binary	Yes	Yes	Yes	Yes	
Char	Yes	Yes	Yes	Yes	
Char for bit data	Yes	Yes	Yes	Yes	
Date	Yes	Yes	Yes	Yes	
Decimal	Yes	Yes	Yes	Yes	
Double	Yes	Yes	Yes	Yes	
Graphic	Yes	Yes	Yes	Yes	
Integer	Yes	Yes	Yes	Yes	You must enter a precision value up to 10.
Long Varchar	Yes	Yes	Yes	Yes	

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Comments
Long Varchar for bit data	Yes	Yes	Yes	Yes	
Long Vargraphic	Yes	No	Yes	Yes	
Numeric	Yes	Yes	Yes	Yes	
Real	Yes	Yes	Yes	Yes	
Smallint	Yes	Yes	Yes	Yes	You must enter a precision value up to 19.
Time	Yes	Yes	Yes	Yes	
Timestamp	Yes	Yes	Yes	Yes	
Varbinary	Yes	Yes	Yes	Yes	
Varchar	Yes	Yes	Yes	Yes	
Varchar for Bit Data	Yes	Yes	Yes	Yes	
Vargraphic	Yes	Yes	Yes	Yes	

When you want to generate test data, you must restrict all the above mentioned data types that are not supported. You cannot perform TDM operations on the following DB2 for IOS data types:

- Blob
- Clob
- Datalink
- Dbclob
- Clob
- Datalink
- Decfloat
- Rowid
- User_defined

IMS Flat

The following table describes the operations that you can perform on data types for IMS Flat file system:

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Bin	Yes	Yes	No	Yes	Yes	You cannot run entity profiles.
Char	Yes	Yes	Yes	Yes	Yes	
Date	Yes	Yes	Yes	Yes	Yes	
Double	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Float	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num16	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num16U	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num32	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num32U	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num64	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num64U	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num8	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Num8U	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Numchar	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Packed	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Spacked	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Time	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Timestamp	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Upacked	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Uzoned	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Varbin	Yes	Yes	No	Yes	Yes	You cannot run entity profiles.
Varchar	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Zoned	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.

Sybase IQ

The following table describes the operations that you can perform on data types for Sybase IQ database:

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Bigint	Yes	Yes	Yes	Yes	Yes	
Binary	Yes	Yes	Yes	Yes	No	You can assign the nullification data masking rule.
Bit	Yes	Yes	Yes	Yes	Yes	You can enter the following entity criteria: Is Null and Is Not Null.
Blob	Yes	Yes	Yes	Yes	No	You can assign the nullification data masking rule.
Char	Yes	Yes	Yes	Yes	Yes	
Clob	Yes	Yes	Yes	Yes	No	
Date	Yes	Yes	Yes	Yes	Yes	

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Datetime	Yes	Yes	Yes	Yes	Yes	
Decimal	Yes	Yes	Yes	Yes	Yes	You can enter precision value up to 28. You cannot run entity profiles.
Double	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Float	Yes	Yes	Yes	Yes	Yes	You can enter precision value up to 28. You cannot run entity profiles.
Image	Yes	Yes	Yes	Yes	No	You can assign the nullification data masking rule.
Int	Yes	Yes	Yes	Yes	Yes	
Long Binary	Yes	Yes	Yes	Yes	No	You can assign the nullification data masking rule.
Long Varchar	Yes	Yes	Yes	Yes	No	
Money	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Numeric	Yes	Yes	Yes	Yes	Yes	You can enter precision value up to 28. You cannot run entity profiles.
Real	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Smalldatetime	Yes	Yes	Yes	Yes	Yes	
Smallint	Yes	Yes	Yes	Yes	Yes	
Smallmoney	Yes	Yes	Yes	Yes	Yes	You cannot run entity profiles.
Sysname	Yes	Yes	Yes	Yes	Yes	

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Profiling	Comments
Text	Yes	Yes	Yes	Yes	No	
Time	Yes	Yes	Yes	Yes	Yes	
Timestamp	Yes	Yes	Yes	Yes	Yes	
Tinyint	Yes	Yes	Yes	Yes	Yes	When you assign a data generation rule, ensure that you provide a range between 0 and 255.
Unique Identifier	No	Yes	No	Yes	Yes	You can enter the Is Null entity criteria.
Unsigned bigint	Yes	Yes	Yes	Yes	Yes	
Unsigned int	Yes	Yes	Yes	Yes	Yes	
Unsigned tinyint	Yes	Yes	Yes	Yes	Yes	
Varbinary	Yes	Yes	Yes	Yes	No	You can assign the nullification data masking rule.
Varchar	Yes	Yes	Yes	Yes	Yes	

Netezza

The following table describes the operations that you can perform on data types in Netezza database:

Data Types	Data Generation
Bigint	Yes
Boolean	Yes
Bpchar	Yes
Byteint	Yes
Char	Yes

Data Types	Data Generation
Date	Yes
Double	Yes
Float	Yes
Int1	Yes
Int2	Yes
Int4	Yes
Int8	Yes
Integer	Yes
Nchar	Yes
Nvarchar	Yes
Real	Yes
Smallint	Yes
Time	Yes
Timestamp	Yes
Varchar	Yes

You cannot perform data masking, data subset, profiling, and data generation operations on the following Netezza data types:

- Decimal
- Decimal (38,0)
- Decimal (38,38)
- Interval
- Interval day
- Interval day to hour
- Interval day to minute
- Interval hour to second
- Interval hour
- Interval hour to minute
- Interval hour to second
- Interval minute
- Interval minute to second
- Interval month
- Interval second

- Interval year
- Interval year to month
- Numeric
- ST_Geometry
- Timetz
- Varbinary

Teradata

The following table describes the operations that you can perform on data types in Teradata database:

Data Types	Data Generation	Comments
Bigint	Yes	
Blob	Yes	You must set the precision in the range of 1 to 64000.
Byte	Yes	
Byteint	Yes	
Char	Yes	
Clob	Yes	You must set the precision in the range of 1 to 64000.
Date	Yes	
Decimal	Yes	You must set the precision in the range of 1 to 37.
Double precision	Yes	
Float	Yes	
Integer	Yes	
Interval day	Yes	
Interval day to hour	Yes	
Interval day to minute	Yes	
Interval day to second	Yes	
Interval hour	Yes	
Interval hour to minute	Yes	

Data Types	Data Generation	Comments
Interval hour to second	Yes	
Interval minute	Yes	
Interval minute to second	Yes	
Interval month	Yes	
Interval second	Yes	
Interval year	Yes	
Interval year to month	Yes	
Long varchar	Yes	
Real	Yes	
Smallint	Yes	
Time	Yes	
Time with time zone	Yes	You must set the precision in the range of 1 to 29.
Timestamp	Yes	
Timestamp with time zone	Yes	
Varbyte	Yes	
Varchar	Yes	

You cannot perform data masking, data subset, profiling, and data generation operations on the following Teradata data types:

- Graphic
- Long vargraphic
- Mbr
- Period (date)
- Period (time with time zone)
- Period (time)
- Period (timestamp with time zone)
- Period (timestamp)
- St_geometry
- Vargraphic

Cassandra

The following table describes the data masking and data domain discovery operations that you can perform on the Cassandra data types:

Data Types	Data Masking	Data Domain Discovery
ASCII	Yes	Yes
Bigint	Yes	Yes
Blob	Yes	Yes
Boolean	Yes	Yes
Decimal	Yes	Yes
Double	Yes	Yes
Float	Yes	Yes
Inet	Yes	Yes
Int	Yes	Yes
Text	Yes	Yes
Timestamp	Yes	Yes
UUID	Yes	Yes
TimeUUID	Yes	Yes
Varchar	Yes	Yes
Varint	Yes	Yes

You cannot perform data masking and data domain discovery operations on the following Cassandra data types:

- Counter
- Custom

You cannot perform an inplace masking operation on a primary key column for Cassandra data sources.

MongoDB

The following table describes the data masking, data subset, data discovery, and data generation operations that you can perform on the MongoDB data types:

Data Types	Data Movement	Entity Criteria	Data Masking	Data Generation	Data Domain Discovery	Comments
Binary	Yes	No	Yes	No	No	You cannot perform data domain discovery and conditional data masking operations. You cannot preview data for a Binary column in the data domain profile results.
Boolean	Yes	Yes	Yes	Yes	Yes	
Date	Yes	Yes	Yes	Yes	Yes	
jstOID	Yes	Yes	Yes	Yes	Yes	
Number Double	Yes	Yes	Yes	Yes	Yes	You can enter a precision value up to 15.
NumberInt	Yes	Yes	Yes	Yes	Yes	You can enter a precision value up to 10.
NumberLong	Yes	Yes	Yes	Yes	Yes	You can enter a precision value from 1 to 28. You can a scale value from 0 to 28.
String	Yes	Yes	Yes	Yes	Yes	

You can use the following special characters for MongoDB sources: `/ + - ~ ? , | @ #`

Use the following driver version for MongoDB:

```
SimbdaMongoDBODBC Driver Version: 1.8.x.xxx
```

You cannot perform the data masking, data subset, data discovery, and data generation operations on the following MongoDB data type:

- Javascript

- Timestamp

APPENDIX B

Data Type Reference for Test Data Warehouse

This appendix includes the following topics:

- Data Type Reference for Test Data Warehouse Overview, 370
- Oracle, 371
- Microsoft SQL Server, 373
- DB2 for Linux, UNIX, and Windows, 375
- DB2 for z/OS, 376
- Sybase ASE, 377
- Teradata, 379
- MongoDB, 381

Data Type Reference for Test Data Warehouse Overview

You can perform data subset, data masking, and data generation operations and store the results in a test data warehouse.

When the target is test data warehouse, you can perform the TDM operations on the supported data types in the following databases:

- Oracle
- Microsoft SQL Server
- DB2 for Linux, UNIX, and Windows
- Sybase ASE
- Teradata

Oracle

The following table describes the data masking, data subset, and data generation operations that you can perform on the Oracle data types when the target is test data warehouse:

Data Types	Test Data Warehouse	Comments
Binary_Double	Yes	You can enter a precision value up to 28.
Binary_Float	Yes	
Blob	Yes	
Char	Yes	
Clob	Yes	
Date	Yes	
Decimal(P,S)	Yes	You can enter a precision value up to 28. You cannot enter the data type with value Decimal(38,38).
Float	Yes	
Integer	Yes	
Long	Yes	
Longraw	Yes	
Longvarchar	Yes	
Nchar	Yes	
Nclob	Yes	
Number	Yes	You can enter a precision value up to 28. You cannot enter the data type with value Number(38,38).
Nvarchar2(N)	Yes	
Raw	Yes	
Real	Yes	
Smallint	Yes	
Timestamp	Yes	

Data Types	Test Data Warehouse	Comments
Timestamp with local time zone	Yes	You must set the precision to 40. You cannot perform the test generation operation.
Timestamp with time zone	Yes	You must set the precision to 40. You cannot perform the test generation operation.
Urowid	Yes	You cannot assign a data generation rule. TDM generates null values in the target.
Varchar2(N)	Yes	
XML	Yes	If you assign a dictionary generation rule that contains specified invalid values for the XML data type in an Oracle database, the data generation task fails.

When the target is test data warehouse, you cannot perform data masking, data subset, and data generation operations on the following Oracle data types:

- Anydata
- Anydataset
- Anytype
- Bfile
- Dburitype
- Httpuritype
- Interval day to second
- Interval year to month
- Ordaudio
- Orddicom
- Orddoc
- Ordimage
- Ordimagesignature
- Ordvideo
- Rowid
- Sdo_Geometry
- Sdo_Georaster
- Sdo_Topo_Geometry
- Sl_Averagecolor
- Sl_Color

- SI_Colorhistogram
- SI_Featurelist
- SI_Positionalcolor
- SI_Stillimage
- SI_Texture
- Uritype
- Xdburitype

Microsoft SQL Server

The following table describes the data masking, data subset, and data generation operations that you can perform on the Microsoft SQL Server data types when the target is test data warehouse:

Data Types	Test Data Warehouse	Comments
Bigint	Yes	
Binary	Yes	Use dictionary generation rule to generate test data. Use nullification data masking rule to mask the data.
Bit	Yes	
Char	Yes	
Date	Yes	
Datetime	Yes	
Datetime2	Yes	
Datetimeoffset	Yes	
Decimal	Yes	You can enter a precision value up to 28.
Float	Yes	To perform a data masking operation, you must set the precision to less than 28. When you run a data generation operation, TDM truncates data if the precision is greater than 17.
Image	Yes	
Int	Yes	
Money	Yes	
Nchar	Yes	

Data Types	Test Data Warehouse	Comments
Ntext	Yes	To perform a reset operation on tables with Ntext data type, create a logical primary key in Test Data Manager if a physical primary key does not exist.
Numeric	Yes	You can enter a precision value up to 28.
Nvarchar	Yes	
Real	Yes	When you run a data generation operation, TDM truncates data if the precision is greater than 15. To perform a reset operation on tables with Real data type, create a logical primary key in Test Data Manager if a physical primary key does not exist.
Smalldatetime	Yes	
Smallint	Yes	
Smallmoney	Yes	
Sql_variant	Yes	You cannot perform a data masking operation.
Text	Yes	To perform a reset operation on tables with Text data type, create a logical primary key in Test Data Manager if a physical primary key does not exist.
Time	Yes	When you enter values in milliseconds, TDM truncates the data.
Tinyint	Yes	
Uniqueidentifier	Yes	You cannot assign string masking rule even if the data type is present in the `dataMapping.xml` file. You cannot perform data generation operations.
Varbinary	Yes	Use dictionary generation rule to generate test data. Use nullification data masking rule to mask the data.
Varchar	Yes	
XML	Yes	You cannot perform a data masking operation.

When the target is test data warehouse, you cannot perform data masking, data subset, and data generation operations on the following Microsoft SQL Server data types:

- Geography
- Geometry
- HierarchyID
- Timestamp

DB2 for Linux, UNIX, and Windows

The following table describes the data masking, data subset, and data generation operations that you can perform on the DB2 data types when the target is test data warehouse:

Data Types	Test Data Warehouse	Comments
Bigint	Yes	
Blob (N)	Yes	
Char for bit data	Yes	You can perform the data generation operation.
Char varying (Length)	Yes	
Character (N)	Yes	
Clob (N)	Yes	
Date	Yes	
Dbclob (N)	Yes	
Dec	Yes	You can perform the data generation operation.
Decfloat (16)	Yes	
Decfloat (34)	Yes	
Decimal	Yes	
Double	Yes	
Double Precision	Yes	
Float	Yes	
Graphic (N)	Yes	
Integer	Yes	
Long Varchar	Yes	
Long Vargraphic	Yes	
Num	Yes	
Numeric	Yes	
Real	Yes	
Smallint	Yes	
Time	Yes	You cannot assign a data generation rule. TDM generates null values in the target.

Data Types	Test Data Warehouse	Comments
Timestamp	Yes	
Varchar for Bit Data	Yes	
Varchar (N)	Yes	
Vargraphic (N)	Yes	
XML	Yes	Use dictionary generation rule to generate data. You cannot perform data masking operations.

DB2 for z/OS

The following table describes the data masking, data subset, and data generation operations that you can perform on the DB2 for z/OS data types when the target is test data warehouse:

Data Types	Test Data Warehouse	Comments
Bigint	Yes	
Binary	Yes	
Char	Yes	
Char for bit data	Yes	
Date	Yes	
Decimal	Yes	
Double	Yes	
Float	Yes	
Integer	Yes	
Long Varchar	Yes	
Long Varchar for bit data	Yes	
Real	Yes	
Smallint	Yes	
Time	Yes	
Timestamp	Yes	

Data Types	Test Data Warehouse	Comments
Varbinary	Yes	
Varchar	Yes	
Varchar for Bit Data	Yes	

When the target is test data warehouse, you cannot perform data masking, data subset, and data generation operations on the following DB2 for z/OS data types:

- Blob
- Clob
- Dbclob
- Decfloat
- Graphic
- Long Vargraphic
- Vargraphic
- XML

Sybase ASE

The following table describes the data masking, data subset, and data generation operations that you can perform on the Sybase ASE data types when the target is test data warehouse:

Data Types	Test Data Warehouse	Comments
Bigdatetime	Yes	
Bigint	Yes	
Bigtime	Yes	
Bit	Yes	
Binary (n)	Yes	
Char (n)	Yes	
Date	Yes	
Datetime	Yes	
Decimal (P,S)	Yes	You can enter a precision value up to 28. You cannot enter the data type with value Decimal(38,38).

Data Types	Test Data Warehouse	Comments
Double Precision	Yes	
Float (P)	Yes	
Image	Yes	
Int	Yes	
Longsysname	Yes	
Money	Yes	
Nchar (N)	Yes	
Numeric (P,S)	Yes	You can enter a precision value up to 28. You cannot enter the data type with value Numeric(38,38).
Nvarchar (N)	Yes	
Real	Yes	
Smalldatetime	Yes	
Smallint	Yes	
Smallmoney	Yes	
Sysname	Yes	
Text	Yes	
Time	Yes	
Tinyint	Yes	When you assign a data generation rule, ensure that you provide a range between 0 and 255.
Unitext	Yes	
Unsigned bigint	Yes	You cannot provide negative values.
Unsigned int	Yes	You cannot provide negative values.
Unsigned smallint	Yes	You cannot provide negative values.
Unsigned tinyint	Yes	When you assign a data generation rule, ensure that you provide a range between 0 and 255.
Varbinary (N)	Yes	
Varchar (N)	Yes	

When the target is test data warehouse, you cannot perform data masking, data subset, and data generation operations on the following Sybase ASE data types:

- Timestamp. You cannot perform a data masking operation on Timestamp data type because the value is system generated in Sybase database.
- Unichar
- Univarchar

Teradata

The following table describes the operations that you can perform on data types in a Teradata database:

Data Types	Test Data Warehouse	Comments
Bigint	Yes	
Blob	Yes	You must set the precision in the range of 1 to 4000.
Byte	Yes	
Byteint	Yes	
Char	Yes	
Char varying	Yes	
Clob	Yes	You must set the precision in the range of 1 to 4000.
Date	Yes	
Decimal	Yes	You must set the precision in the range of 1 to 38.
Double precision	Yes	
Float	Yes	
Integer	Yes	
Interval day	Yes	
Interval day to hour	Yes	
Interval day to minute	Yes	
Interval day to second	Yes	
Interval hour	Yes	
Interval hour to minute	Yes	

Data Types	Test Data Warehouse	Comments
Interval hour to second	Yes	
Interval minute	Yes	
Interval minute to second	Yes	
Interval month	Yes	
Interval second	Yes	
Interval year	Yes	
Interval year to month	Yes	
Long varchar	Yes	
Number	Yes	
Real	Yes	
Smallint	Yes	
Time	Yes	
Time with time zone	Yes	You must set the precision in the range of 1 to 29.
Timestamp	Yes	
Timestamp with time zone	Yes	
Varbyte	Yes	
Varchar	Yes	

You cannot perform test data warehouse operations on the following Teradata data types:

- Graphic
- Long vargraphic
- Mbr
- Period (date)
- Period (time with time zone)
- Period (time)
- Period (timestamp with time zone)
- Period (timestamp)
- St_geometry
- Vargraphic

MongoDB

The following table describes the data masking, data subset, and data generation operations that you can perform on the MongoDB data types when the target is test data warehouse:

Data Types	Test Data Warehouse	Comments
Binary	Yes	You cannot apply entity criteria on the Binary data type.
Boolean	Yes	
Date	Yes	
jstOID	Yes	
NumberDouble	Yes	
NumberInt	Yes	
NumberLong	Yes	
String	Yes	

When the target is test data warehouse, you cannot perform data masking, data subset, and data generation operations on the following MongoDB data type:

- Timestamp

APPENDIX C

Glossary

accelerator

An Informatica solution that contains predefined data subset and data masking components for major business applications.

ad hoc generation rule

A data generation rule that you can create within a project. You can use ad hoc generation techniques within an ad hoc generation rule.

advanced generation rule

A data generation rule that is a combination of generation techniques that generates test data in target columns based on the configuration of one or more than one input port, variable port, and output port.

advanced masking

A data masking technique that can include other masking techniques and multiple input and output ports.

auto cascade

A cascade type that masks the primary key in the parent table and the related foreign keys in the child tables with the same value.

blurring

A masking parameter for numeric and date values that creates an output value within a fixed or percent variance from the source data value. Configure blurring to return a random value that is close to the original value.

calculated masking

A masking technique that uses an expression to combine the values of multiple ports.

cascade

A data masking component that masks columns across tables based on column value. Define similar columns in Test Data Manager in order to mask them with the same rules.

column profile

A type of data analysis that determines the characteristics of columns in a data set, such as value frequency, percentages, and patterns.

conditional constraint

A constraint to restrict the data that you want to generate in a column based on the data in another column.

conditional generation rule

An ad hoc generation rule in which you can specify a conditional expression and a generation rule to generate test data.

custom generation rule

A data generation rule that uses PowerCenter mapplets to generate data.

data coverage task

A task that you create to analyze combinations of data values in a data set for density of data coverage. The task generates a graphical representation of data combinations plotted along the X axis and Y axis.

data discovery

The process of discovering the metadata of source systems that includes content, such as data values and frequencies, and structure such as primary keys, foreign keys, and functional dependencies.

data generation

The process to generate realistic test data for the testing environment without using the production data.

data masking

The process of replacing sensitive columns of source data with realistic test data.

data pack

Test data that you store in the test data warehouse and publish to the TDM Self-Service Portal. Users who typically use TDM to access test data access and work with the test data from the self-service portal.

data set

Test data that you store in the test data warehouse. Contains a collection of data that consists of tables from one or more data sources.

data subset

A small, targeted, and referentially intact copy of production data.

dependent masking

A masking technique that replaces the values of one source column based on the values of another source column.

dictionary

A flat file or relational table of data that substitution masking uses to replace a column of data with similar but unrelated data.

dictionary generation

A standard data generation technique that imports dictionary values into the target column.

effective dates generation

An ad hoc generation technique that generates related dates for two columns in the target database.

encryption masking

A data masking technique that applies encryption algorithms to mask source data. You can choose the algorithm to encrypt the data.

entity

Defines a set of tables that are related based on physical or logical constraints. An entity can contain parent tables and child tables. An entity maintains relational hierarchy in the tables. Define filter parameters for ports in the entity to extract the columns from the tables in an entity. When you run a workflow generated from a data subset plan, the PowerCenter Integration Service extracts source data based on the filter parameters defined in the entity and loads the data to the subset tables.

Expression Builder

A wizard in Test Data Manager that includes Informatica functions, variables, and operators to build expressions from the input and the output column values.

expression generation

An ad hoc generation technique that generates data based on the expression.

expression masking

A data masking technique that applies an expression to a port to change or create data.

foreign key profile

A type of data analysis that finds column values in a data source that match the primary key column values in another data source.

generation rule

A rule that you create to define how you want to generate data in the target table.

group

Defines a set of tables to copy to a target subset database.

ilmcmd

A command line program that you can use to complete TDM tasks from any machine. ilmcmd allows you to complete tasks that you can complete in Test Data Manager.

key masking

A data masking technique that produces deterministic results for the same source data, masking rules, and seed value.

mask format

A masking parameter that limits each character in the output column to an alphabetic, numeric, or alphanumeric character.

nullification masking

A data masking technique that replaces column data with null values.

pass phrase

A key that you use in a data masking plan that includes the encryption masking technique with the format preserving encryption type. The pass phrase generates a key to encrypt or decrypt data.

pattern

A data masking component that uses a regular expression to filter the ports that you want to mask when you assign a rule to columns. Use a pattern to mask sensitive data such as IDs, telephone numbers, postal codes, and state names.

plan

Defines data subset, data masking, or data generation operations. You can add entities, groups, templates, policies, rules, and tables to a plan. When you generate and run workflows from a plan, the PowerCenter Integration Service generates and runs the workflows.

policy

Defines the data masking rules, the data to mask, and masking parameters for a source.

PowerCenter

An Informatica solution that can extract data from multiple sources, transform the data according to business logic you build in the client application, and load the transformed data into file and relational targets for data warehousing.

profile

Used in data discovery.

Defines the tables, fields, and keys to discover. You can create profiles to discover the following items in tables:

- Sensitive fields
- Foreign keys
- Primary keys

project

A container component for entities, groups, templates, and one or more sources that you want to use in data discovery, subset, and masking operations. When you create a project, you add one or more sources to the project. Any source that you add to a project is not available to other projects.

random generation

A standard data generation technique that generates random data for string, numeric, and date datatypes.

random masking

A masking technique that produces random, non-repeatable results for the source data and masking rules.

range

A data masking parameter that defines the minimum and maximum values to mask the numeric and date datatypes, and the minimum and maximum values for string lengths of string datatype.

reference lookup generation

An ad hoc generation technique that generates data from a reference table.

related data set

A data subset that you create from a data set. A related data set contains a specific chunk of data from a data set that is referentially intact.

residual table

In a data subset plan, a table that is missing from entities and groups in the plan but which is part of the project. You can add residual tables to a data subset plan.

result string replacement characters

A data masking parameter that defines the substitute characters for masked data.

rule

Defines the data masking technique, an optional rule qualifier, and masking parameters.

rule qualifier

A rule component that filters the records that you want to mask.

seed

A start number that key masking uses to return deterministic data.

sequence generation

A standard data generation technique that generates numeric and date values in a sequence.

set of values generation

A standard data generation technique that defines a finite set of values to write to the target table.

shuffle masking

A data masking technique that masks the data in a column with data from the same column from another row of the table.

simulation masking

A masking type that generates a simulation report without masking any data. The report contains policies, rules, masking parameters, qualified ports, and column values that should be masked and their corresponding masked values.

source string characters

A data masking technique that masks the selected characters. You can choose to mask the selected source string characters or mask all characters except the source string characters.

standard generation rule

A data generation rule that applies a predefined data generation technique.

substitution masking

A data masking technique that replaces a column of data with similar but unrelated data.

TDM repository

A relational database that stores the components that you define in Test Data Manager, such as policies, projects, entities, and data masking rules. The TDM repository stores metadata that you import into Test Data Manager from a source database or from the PowerCenter repository. The TDM repository stores constraints that define relationships between the source tables in a project.

TDM Server

Server that runs Test Data Manager and integrates with Informatica application services to perform data subset, data masking, and data discovery operations.

tdwcmd

A command line program that you use to perform test data warehouse tasks. You can create a test data warehouse repository, create a test data warehouse, and list all the test data warehouse details.

tdwquery

A command line program that you use to analyze data in a data set version. You can run queries with multiple joins and clauses to filter and search for data.

template

Used in data subset.

A reusable component that contains the entities and groups of a specific module that you want to subset based on common parameters.

Test Data Management (TDM)

The Informatica solution that bundles Data Subset, Data Generation, and Data Masking to protect sensitive data and create lean nonproduction systems for test and development purposes.

Test Data Management Self-Service Portal

A Test Data Management portal that provides simplified access to test data stored in the test data warehouse. Users who work with test data in TDM but do not create the test data access the data from the self-service portal.

Test Data Manager

The web-based user interface that you use to configure and run data subset, data masking, and discovery operations.

test data repository

A relational database that stores the project metadata and the source table metadata related to data sets that you store in the test data warehouse.

test data warehouse

A central and location to store and manage test data in TDM. You can store data from data subset, data masking, and data generation plans. Restore or reset required data from the test data warehouse to a target connection in TDM.

value cascade

A cascade type that masks the column values in different tables with the same value.

INDEX

A

abort
  workflows 297
Actions menu
  Test Data Manager 37
ad hoc
  conditional 225
  generation plan 291
ad hoc generation rules
  creating 188
  description 188
  editing 189
Administrator view
  description 44
advanced generation
  assign 195
  example 203
  parameters 205
advanced masking
  description 148
  parameters 149
advanced options
  plan settings 275
advanced rules
  assigning rule to columns 134
  creating 133
  description 129
advanced search
  text match 38
advanced settings
  post workflow 268
  pre workflow 268
application services
  TDM 26
architecture
  TDM 24
asset linking
  create link 76
  delete link 77
assign
  advanced 135
  advanced generation rule 195
    advanced rule
      assignment 135
  custom generation 195
  mapplet rule 134
assign masking rule
  xml 135
assigning
  rules in policies to columns in an project source 134
assignments
  deleting 140
  multiple rules to a column 139
  overriding a rule 138
  rules to columns 130, 193
auto cascades
  overview 85

B

blurring
  random masking 160

C

calculated masking
  description 150
cascades
  auto 85
  value 84
CLI
  ilmcmd description 311
column profile
  profiled patterns 99
  results 99
  inferred rules 90
  list of values 90
  patterns 90
  range 90
  sampling options 91
columns
  cascades 83
  computed 83
  sensitivity 83
commit interval
  workflow properties 274
commit type
  workflow properties 272, 274
components
  data generation 182
  data masking 141
conditional
  constraints 192
conditional constraint
  create 104
connections
  Administrator view 44
  configuring workflows 267
  plan settings 269
  TDM 28
  to databases 29
  to repositories 29
  to services 28
constraint
  major and minor severity 102
    relationships
      optimize 113
constraint properties
  entity map 111

constraint type
conditional 104
constraint-based loading
workflow properties 272, 274
constraints
adding 103
reviewing changes 114
content panel
Test Data Manager 37
copying
data set plans 243
plans 292
creating
conditional constraints 104
entities 110
credit card generation
issuer identification number 218
parameters 219
credit card masking
description 150
criteria
data subset 281
parameters 111
subset and masking 281
custom generation
assign 195
parameters 203
custom generation rules
description 187
custom mapplet
masking rule 148
custom masking
parameters 148

D

dashboards
Test Data Manager 40
data coverage
analysis 251
column types 252
move data 259
data coverage column
mapping 256
range 255
use as is 255
data coverage task
edit 258
data discovery
column profile 90
nonrelational sources 81
overview 80
rules and guidelines 81
task flow 85
TDM process 31
data discovery profiles
creating 93
deleting 95
editing 94
data domain
default masking rules 67
delete 71
edit metadata 70
data domain assignments
exporting and importing 90
exporting from Secure@Source 89
data domain definitions
data patterns 69
metadata patterns 69
data domain profile
sampling options in 88
data domains
assigning to multiple columns 89
copying 70
creating 70
data generation rules 67
discovering 88
exportingand importing 74
manually adding to columns 89
options 69
overview 67
status 69
data expressions
patterns 69
data generation
expression generation 211
random generation 214
sequence generation 223
set of values 224
ad hoc rules 188
assign auto rule 196
components 182, 281
conditional constraints 192
creating ad hoc rules 188
creating standard rules 186
credit card 217
custom rules 187
deleting rules 187
dictionary generation 207
editing ad hoc rules 189
editing rules 186
effective dates generation
parameters 209
exception test data 201
overview 181
parameters 201
plans 198
rules 199
standard rules 185
task flow 183
techniques 199, 200
data generation parameters
random generation 212, 213
reference lookup 221
data generation rules
preferred rules 67
XML elements and attributes 197
data masking
assigning multiple rules to a column 139
assigning rules in policies to columns 134
components 140, 141
creating advanced rules 133
creating rules from a mapplet 132
creating standard rules 131
delete rules 138
disabling at the plan level 282
editing rules 137
masking techniques list 144
metadata and data patterns 68
overview 127
plan 140
plans 265
repeatable output 146
rules 144

data masking *(continued)*
rules overview 128
standard rules 129
task flow 128
techniques 144
data masking plans
creating 290
data patterns
data domain definitions 69
TDM data domains 68
data set
edit 238
classify tables 232
create 230
creating groups 241
delete 239
entities 240
lock 245
metadata 238
related 240
tags 238
tasks 228
unlock 245
data set version
delete 239
Data sets
subset 239
data sources
Hadoop HDFS 57
xsd 57
data subset
create 118
creating groups 115
editing groups 116
entities 110, 116
overview 106
plans 265
process 31
data subset plans
creating 290
databases
connections to 29
TDM 27
date format string
workflow properties 274
date values
key masking 158
random masking 162
default rule
data domain 185
TDM data domains 67
Define view
Test Data Manager 48
definitions
TDM repository 57
delete
ilmcmd command 314
delete table
impacted objects 62
project 62
deleting 293
deleting a table
project 62
details panel
Test Data Manager 37
dictionary connections
plan settings 269
relational and flat file 270
dictionary generation
description 206
parameters 207
dictionary location
flat files 270
discover
columns 83
discover
profiles 82
tables 82
Discover view)
Test Data Manager 48
discovering
data domains 88
primary key 86
domains
assigning to multiple columns 89
exporting and importing 74
duplicate
data domains 70
entities 118
groups 118
masking rules 138
policies 78
projects 54

E

edit test data
add rows 248
delete rows 248
row-level tag 249
SQL 247
tag 249
editing
entities 116
effective dates generation
description 208
email masking
description 151
parameters 151
encryption masking
description 153
parameters 154
entities
circular relationships 108
copying 118
creating 110
deleting 118
editing 116
exporting TDM 117
importing TDM 117
overview 108
entity
TDM projects 49
entity discovery
description 80
entity map
parent child ratio 111
entity profiles
inference options 87
example
data subset with XSD source 119
examples
TDM 33
export
ilmcmd command 315

exporting
entities 117
groups 117
policy 74
TDM projects 56
expression
cascade 135
Expression Builder
overview 45
expression generation
parameters 211
description 210
expression masking
description 155
parameters 155
rules and guidelines for 155

F

filter criteria
disabling masking by column 282
filtering data
Test Data Manager 39
foreign key
constraints 103
format preserving encryption 153

G

generate staging DDL
workflow properties 274
generating
workflows 198
generation rule
advanced 190
generation rules
advanced 203
assigning 194
deleting 187
description 183
editing 186
generation techniques
dictionary 206
effective dates 208
expression 210
random 211
reference lookup 220
sequence 221
set of values 223
groups
copying 118
creating 115
deleting 118
editing 116
exporting TDM 117
importing TDM 117
overview 110

H

Hadoop HDFS
PowerCenter mappings 57
high availability recovery
workflow properties 274
high precision
workflow properties 274

I

ilmcmd
command line help 312
ListPlans 322
reset 320
TDWPlanGenExe 324
ilmcmd commands
delete 314
entering options and arguments 313
export 315
import 316
overview 311
search 317
syntax for 313
workflow 319
import
ilmcmd command 316
policy 75
rule 75
importing
entities 117
groups 117
metadata from PowerCenter into TDM 60
plans 292
profiles 96
TDM projects 56
importing metadata
from PowerCenter into Test Data Manager 60
index
disabling in plan 275
inference options
entity profiles 87
inferred rule
types 90
integers
masking large integers 137
IP address masking
description 156

K

key masking
date values 158
description 156
masking numeric values 158
masking string values 158
result string replacement characters 157
source string characters 157

L

limit
data subset plan 281
locale
plan settings 275
logging in
Test Data Manager 46
logs
project 51

M

major severity
  configuring constraints 102
manage logs
  project 55
mandatory columns
  description 133
mapplet
  parameters 148, 203
mapplets
  assigning rule to columns 134
  creating rules from 129
  TDM 142
mark changes review complete
  entity changes 114
mask format
  description 156
masking
  disabling by column 281
  large integers 137
masking component
  policies 140
  rules 140
masking options
  blurring 160
  mask format 156
  range 160
  result string replacement characters 157
  seed 146
  source string characters 157
masking parameters
  advanced masking 149
  date key masking 158
  date random masking 162
  email masking 151
  expression masking 155
  numeric key masking 158
  numeric random masking 163
  random strings 163
  string key masking 158
  substitution 167
masking rule
  copying 138
  custom mapplet 148
  mapplet 134
  xml attribute 135
  xml element 135
masking rules
  delete 136, 138
  editing 137
  update 136
masking techniques
  advanced 148
  calculated 150
  credit card 150
  email 151
  encryption 153
  expression 155
  IP address 156
  key masking 156
  lists 144
  nullification 159
  phone 159
  random 159
  Social Security number 165
  substitution 166
  URL 167
masking types
  format preserving encryption masking 153
  standard masking 153
maximum confidence percent
  inference options 87
maximum foreign keys returned
  inference options 87
metadata patterns
  data domain definitions 69
  TDM data domains 68
minimum confidence percent
  inference options 87
minor severity
  configuring constraints 102
Monitor
  details 302
  jobs 301–303
  logs 303
  tasks 302
Monitor view
  description 300
  Test Data Manager 44

N

nullification masking
  description 159
numeric random generation
  parameters 214
numeric sequence generation
  parameters 223
numeric values
  key masking 158
  random masking 163

O

overriding a rule
  data masking 138
overview
  data discovery 80
  data generation 181
  data masking 127
  groups 110
  TDM 20
Overview view
  Test Data Manager 40

P

parameter
  create 54
  delete 55
  edit 55
  parameter file 295
  parameter values 295
parameters
  connection 54
  criteria 54
  data generation 199, 201
  data masking 144
  date sequence generation 222
  owner 54
partitioning
  properties 288

patterns
data domain definitions 69
TDM data domains 68
permission view
data set 244
projects 63
Persistent Data Masking
masking Social Insurance numbers 165
phone masking
description 159
plan criteria
configuring 281
plan settings
advanced options 275
connections 269
description 268
source settings 282
Workflow Executions view 297
Workflow view 294
plans
creating 290
data generation 198
deleting 293
disabling masking for columns 282
generating PowerCenter workflows from 293
importing TDM 292
properties 267
TDM projects 49
TDM workflows 265
policies
copying 78
creating 77
editing 78
exporting and importing 74
managing 77
overview 65
task flow 66
Policies view
Test Data Manager 42
policy
delete 78
export 74
PowerCenter
objects 142
PowerCenter repository
importing metadata from 60
PowerCenter workflows
generating from plans 293
PowerExchange sources
importing 60
preferred masking rules
overview 67
primary key
create constraint 103
primary key discovery
description 80
primary key options
primary key profiles 86
primary key profiles
description 86
inference options 86
results 97
processes
data subset 31
profile
TDM projects 49
profiles
data domains 88
profiles *(continued)*
deleting 95
editing 94
importing 96
primary key and entity discovery 86
project
list of tables 100
parameters 50
session properties 51
project risk analysis
description 40
project security
maintaining in TDM 63
projects
copying 54
creating in TDM 52
deleting in TDM 54
editing in TDM 53
exporting TDM 56
importing TDM 56
TDM 48
TDM components 49
Projects view
Test Data Manager 43
properties
plans 267
properties panel
workflow executions 298

Q

quick links
Overview view 40

R

random generation
numeric random 214
description 211
random masking
blurring 160
description 159
masking date values 162
masking string values 163
numeric values 163
range 160
result string replacement characters 157
source string characters 157
range
random masking 160
reference lookup generation
description 220
parameters 221
regenerate signature
inference options 87
regular expressions
syntax 68
related data set
editing entities 241
editing groups 242
related data sets
create 242
subset data set 239
repeatable output
data masking 146

reports
audit trail 307
Test Data Manager 306
repositories
connections to 29
result string replacement characters
key masking 157
random masking 157
results
primary key profiles 97
resume from last checkpoint
workflow properties 274
review changes
entity constraints 114
rollback transactions on error
workflow properties 274
rule assignment
multiple rules to a column 139
rule assignments
description 130, 193
overriding the rule 138
rule simulation
preview rule output 132
simulator 129
standard rule 129
rules
ad hoc generation rules 188
adding to policies 77
advanced 129
assigning multiple rules to a column 139
assigning to columns 130, 134, 193
creating from a mapplet 132
custom generation rules 187
data generation 66
data masking 66
deleting column assignments 140
exporting and importing 74
from PowerCenter mapplets 129
not available for assignment 137, 194
overview 128
standard generation rules 185
standard masking rules 129
rules and guidelines
expression masking 155

S

sampling options
data domain profile 88
search
ilmcmd command 317
seed 146
sensitivity
for ports 83
sensitivity levels
data domains 69
sequence generation
description 221
numeric 223
services
connections to 28
session
details 298
session log
Monitor 304
session properties
storage encryption key 51
sessions
tab 298
set of values generation
parameters 224
description 223
severity
configuring constraints 102
similar value columns
deleting the group 85
description 84
editing 84
setting 84
SIN numbers
masking Social Insurance numbers 165
source settings
plan settings 282
source string characters
key masking 157
random masking 157
sources
TDM 56
special format masking
Social Insurance numbers 165
special masking techniques
credit card 150
email 151
IP address 156
phone 159
Social Security number 165
URL 167
SSN masking
description 165
standard generation rules
creating 186
description 185
standard masking rules
description 129
standard rules
creating 131
start digit
Social Insurance numbers 165
status
data domains 69
stop on error
workflow properties 274
string values
key masking 158
random data masking 163
subset
relations
optimize 113
subset data set
create 242
substitution masking
description 166
parameters 167
syntax
ilmcmd commands 313

T

table classification
description 101
tables
classification 101
tables per mapping
description 275

tags
  row-level 249
targets
  TDM 56
task
  data coverage 251
TDM
  application services 26
  architecture 24
  connections 28
  database connections 29
  databases 27
  example 33
  overview 20
  process overview 30
  repository connections 29
  service connections 28
  sources 56
  targets 56
  tools 26
TDM processes
  data discovery 31
TDM projects
  components 49
  managing 52
  owner 52
TDM repository
  definitions 57
tdwcmd
  arguments 327
  list 328
  options 327
  overview 326
  run command 326
  syntax notation 327
techniques
  data generation 199, 200
  data masking 144
templates
  TDM projects 49
test data
  edit data 246
  view data 246
Test Data Manager
  Administrator view 44
  filtering data 39
  logging in to 46
  Monitor view 44
  Overview view 40
  Policies view 42
  Projects view 43
  reports 306
  user interface overview 36
  views 37
test data warehouse
  reset 231
  reset a data set 234
    security
      data set 244
  tags 244
thinkmaps
  for policy assignments 306
thinkmaps *(continued)*
  for rule assignments 306
tools
  TDM 26
trim spaces before comparison
  inference options 87

U

unique key
  create constraint 104
URL masking
  description 167

V

value cascades
  deleting 85
  description 84
  editing 84
  setting 84
view test data
  SQL 247

W

workflow
  execute 295
workflow executions
  properties panel 298
Workflow Executions view
  plan settings 297
  Sessions tab 298
  tasks 297
Workflow view
  plan settings 294
workflows
  aborting 297
  connections for 267
  generating 198
  generating from plans 293
  ilmcmd command 319
  properties in plan settings 272, 274
  recovering 297

X

XSD
  data generation 197
XSD source
  criteria 116
  example 119
  scope 116