DISCLAIMER: Informatica Corporation provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of non-infringement, merchantability, or use for a particular purpose. Informatica Corporation 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.

Part Number: ICS-USG-10000-0001
# Table of Contents

**Preface** ......................................................................... xiii
Informatica Resources. ............................................................ xiii
  Informatica Documentation. ............................................. xiii
  Informatica Web Site. ...................................................... xiii
  Informatica Cloud Web Site. .......................................... xiii
  Informatica Cloud Community. ....................................... xiii
  Informatica Cloud Trust Site. ......................................... xiv
  Informatica Global Customer Support. ................................ xiv

**Chapter 1: Getting Started.** ............................................. 1
Understanding Informatica Cloud. ........................................... 1
  Informatica Cloud Components. ....................................... 1
  Informatica Cloud Services. ........................................... 2
  Informatica Cloud Subscriptions. ..................................... 2
  Informatica Cloud Security. ........................................... 2
Minimum System Requirements. ............................................ 3
Organization Administrator and User Accounts. ....................... 3
  User Login. ........................................................... 3
  Account Passwords. .................................................. 4
  Account Lockouts. .................................................... 4
  License Expiration. ................................................... 4
Understanding the Informatica Cloud Tabs. .............................. 4
Quick Setup. ............................................................. 5
Help, Support, and Subscribe. .............................................. 5
  Online Help. ........................................................ 5
  Support Page. ....................................................... 5
  Informatica Cloud Subscription Options Page. .................... 5
Configuring User Preferences. .............................................. 6
  Viewing User Preferences. .......................................... 6
  Editing User Preferences. .......................................... 6

**Chapter 2: Performing Administrative Tasks.** ...................... 8
Organization Profile. ....................................................... 8
  Viewing Organization Properties. .................................... 8
Configuring Organization Properties and Details. .................... 10
Understanding License Types. ........................................... 13
Organization Security Overview. ........................................ 14
  Security Examples and Tips. ....................................... 15
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rules and Guidelines for Security</td>
<td>15</td>
</tr>
<tr>
<td>Roles</td>
<td>16</td>
</tr>
<tr>
<td>User Groups</td>
<td>17</td>
</tr>
<tr>
<td>Configuring a User Group</td>
<td>17</td>
</tr>
<tr>
<td>Viewing User Group Details</td>
<td>19</td>
</tr>
<tr>
<td>Deleting a User Group</td>
<td>19</td>
</tr>
<tr>
<td>Object-Level Permissions</td>
<td>19</td>
</tr>
<tr>
<td>Configuring Object-Level Permissions</td>
<td>19</td>
</tr>
<tr>
<td>Users</td>
<td>21</td>
</tr>
<tr>
<td>Configuring a User</td>
<td>21</td>
</tr>
<tr>
<td>Viewing User Details</td>
<td>23</td>
</tr>
<tr>
<td>Deleting a User</td>
<td>23</td>
</tr>
<tr>
<td>Migrating Objects</td>
<td>23</td>
</tr>
<tr>
<td>Migration Requirements</td>
<td>24</td>
</tr>
<tr>
<td>Migrating an Object</td>
<td>24</td>
</tr>
<tr>
<td>Viewing the Audit Log</td>
<td>25</td>
</tr>
<tr>
<td>Billing Information</td>
<td>25</td>
</tr>
<tr>
<td>Viewing Billing Information</td>
<td>25</td>
</tr>
<tr>
<td>Editing Billing Information</td>
<td>25</td>
</tr>
<tr>
<td>Organization Hierarchy</td>
<td>27</td>
</tr>
<tr>
<td>Using the Informatica Cloud Administrator</td>
<td>27</td>
</tr>
<tr>
<td>Creating a Sub-Organization</td>
<td>28</td>
</tr>
</tbody>
</table>

**Chapter 3: Configuring Informatica Cloud Components**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informatica Cloud Secure Agent</td>
<td>31</td>
</tr>
<tr>
<td>Machine Requirements for the Secure Agent</td>
<td>31</td>
</tr>
<tr>
<td>Rules and Guidelines for the Secure Agent</td>
<td>32</td>
</tr>
<tr>
<td>Working with the Secure Agent and Secure Agent Manager on Windows</td>
<td>32</td>
</tr>
<tr>
<td>Working with the Secure Agent on Linux</td>
<td>37</td>
</tr>
<tr>
<td>Connections</td>
<td>41</td>
</tr>
<tr>
<td>Understanding FTP/SFTP Connections</td>
<td>42</td>
</tr>
<tr>
<td>Understanding ODBC Connections</td>
<td>43</td>
</tr>
<tr>
<td>Understanding Salesforce Connections</td>
<td>43</td>
</tr>
<tr>
<td>Understanding SAP Connections</td>
<td>43</td>
</tr>
<tr>
<td>Understanding Web Service Connections</td>
<td>43</td>
</tr>
<tr>
<td>Rules and Guidelines for Connections</td>
<td>44</td>
</tr>
<tr>
<td>Understanding the Service URL</td>
<td>46</td>
</tr>
<tr>
<td>Task Operations</td>
<td>46</td>
</tr>
<tr>
<td>Configuring a Connection</td>
<td>47</td>
</tr>
<tr>
<td>Viewing Connection Details</td>
<td>53</td>
</tr>
<tr>
<td>Copying a Connection</td>
<td>53</td>
</tr>
<tr>
<td>Testing a Connection</td>
<td>53</td>
</tr>
<tr>
<td>Deleting a Connection</td>
<td>54</td>
</tr>
</tbody>
</table>
Schedules ................................. 54
Repeat Frequency ................................................. 54
Time Zones and Schedules. ........................................ 55
Daylight Savings Time Changes and Schedules. .......................... 55
Configuring a Schedule. ........................................... 55
Viewing Schedule Details. .......................................... 57
Deleting a Schedule .................................................. 58

Task Flows ....................................................... 58
Configuring a Task Flow. ......................................... 58
Viewing Task Flow Details. ........................................ 59
Running a Task Flow. .............................................. 59
Deleting a Task Flow ................................................ 60

Plug-ins ............................................................ 60
Active and Passive Plug-ins. ....................................... 60
Stored Procedures in Plug-ins. ..................................... 61
Rules and Guidelines for the PowerCenter XML File. ..................... 61
Rules and Guidelines for Using Plug-ins in Tasks. ........................ 62
Configuring a Plug-in. .............................................. 62
Viewing a Plug-in .................................................. 63
Deleting a Plug-in .................................................. 63

Custom Sources ..................................................... 63
Configuring a Custom Source. ..................................... 64
Rules and Guidelines for Configuring a Custom Source. ................. 64
Rules and Guidelines for Using Custom Sources in Tasks. ............... 64
Steps to Create a Custom Source. .................................. 65
Viewing a Custom Source. .......................................... 65
Copying a Custom Source. ......................................... 65
Deleting a Custom Source .......................................... 66

Chapter 4: Data Assessment Service ........................................ 67
Data Assessment Service Overview ...................................... 67
Understanding Quality Metrics ........................................ 68
Address Validation Quality Metric. ................................... 68
Completeness Quality Metric. ....................................... 68
Conformance Quality Metric. ...................................... 68
Duplicates Quality Metric. ......................................... 68
Understanding Scorecards ............................................ 70
Top Issues ......................................................... 70
Field Scores ....................................................... 70
Individual Scores .................................................. 71
Overall Score ...................................................... 71
Trend Chart ......................................................... 71
Rules and Guidelines for Data Assessment Sources ....................... 72
# Configuring a Data Assessment Task

- Step 1. Configure the Source .............................................. 72
- Step 2. Configure Quality Metrics ............................................ 73
- Step 3. Configure Data Filters .............................................. 74
- Step 4. Configure Scorecard Thresholds ....................................... 75
- Step 5. Configure a Schedule .............................................. 75

# Customizing Weights

- Rules and Guidelines for Weights ........................................... 76

# Customizing Field Mappings

- Rules and Guidelines for Field Mappings ....................................... 77

# Viewing Data Assessment Task Details

# Copying a Data Assessment Task

# Running a Data Assessment Task

- Running a Data Assessment Task Manually ..................................... 78
- Running a Data Assessment Task on a Schedule ................................... 79

# Stopping a Data Assessment Task .............................................. 79

# Monitoring a Data Assessment Task ............................................. 79

# Viewing the Scorecard for a Data Assessment Task ................................ 79

# Analyzing Data Assessment Results ............................................. 79

# Deleting a Data Assessment Task .............................................. 80

---

# Chapter 5: Data Loader Service

- Data Loader Service Overview ................................................ 82
- Rules and Guidelines for Data Loader Sources and Targets .......................... 82
  - Rules and Guidelines for Salesforce Sources and Targets ........................... 83
  - Rules and Guidelines for Flat File Sources and Targets ............................... 83
  - Rules and Guidelines for Database Sources and Targets ............................... 83
- Data Loader Prerequisite Tasks ................................................ 84
- Configuring a Data Loader Task ............................................... 84
  - Step 1. Define the Data Loader Task ......................................... 85
  - Step 2. Configure the Source .............................................. 85
  - Step 3. Configure the Target ............................................... 85
  - Step 4. Configure Data Filters .............................................. 86
  - Step 5. Configure the Field Mapping .......................................... 87
  - Step 6. Configure a Schedule .............................................. 88
- Viewing Data Loader Task Details .............................................. 89
- Running a Data Loader Task .................................................. 89
  - Rules and Guidelines for Running a Data Loader Task .............................. 89
  - Running a Data Loader Task Manually ........................................ 89
  - Running a Data Loader Task on a Schedule ..................................... 90
- Stopping a Data Loader Task ................................................. 90
- Monitoring a Data Loader Task ................................................ 90
- Deleting a Data Loader Task .................................................. 90
IDs for Related Salesforce Objects ............................................. 139
Example of Configuring External IDs for Related Objects ..................... 140
Field Expressions ............................................................................ 141
Rules and Guidelines for Validating Field Mappings .............................. 141
Transformation Language Components for Expressions ......................... 142
Expression Syntax ............................................................................ 142
String and Numeric Literals ........................................................... 142
Rules and Guidelines for Expressions .............................................. 143
Adding Comments to Expressions .................................................... 143
Reserved Words .............................................................................. 144
Data Filters ...................................................................................... 145
Simple Data Filters .......................................................................... 145
Advanced Data Filters ..................................................................... 146
Data Filter Operators ....................................................................... 146
Data Filter Variables ........................................................................ 147
Rules and Guidelines for Data Filters ............................................... 147
Email Notification for Tasks ............................................................ 149
Custom Views .................................................................................. 150
Configuring a Custom View ............................................................ 150
Working with a Custom View ........................................................... 156
Deleting a Custom View ................................................................... 156
Rules and Guidelines for Copying Objects .......................................... 156

Chapter 10: Monitoring Jobs ............................................................. 158
Monitoring Jobs Overview ............................................................... 158
Activity Log ...................................................................................... 158
Viewing the Activity Log ................................................................... 159
Viewing Activity Log Details ............................................................ 159
Searching the Activity Log Based on Task Name ................................... 161
Searching the Activity Log Based on Activity Log Details ..................... 161
Activity Monitor ............................................................................... 162
Stopping a Job .................................................................................. 162
Viewing the Activity Monitor in List View .......................................... 162
Viewing the Activity Monitor in Detail View ....................................... 162
Dashboard ....................................................................................... 162
Configuring the Dashboard .............................................................. 163
Viewing the Dashboard .................................................................... 163
Data Files .......................................................................................... 163
Error Log Files ................................................................................ 163
Viewing a Snapshot of the Error Log File ........................................... 164
Viewing the Error Log File ............................................................... 164
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS_NUMBER</td>
<td>239</td>
</tr>
<tr>
<td>IS_SPACES</td>
<td>240</td>
</tr>
<tr>
<td>ISNULL</td>
<td>241</td>
</tr>
<tr>
<td>LAST_DAY</td>
<td>242</td>
</tr>
<tr>
<td>LEAST</td>
<td>243</td>
</tr>
<tr>
<td>LENGTH</td>
<td>243</td>
</tr>
<tr>
<td>LN</td>
<td>244</td>
</tr>
<tr>
<td>LOG</td>
<td>245</td>
</tr>
<tr>
<td>LOWER</td>
<td>245</td>
</tr>
<tr>
<td>LPAD</td>
<td>246</td>
</tr>
<tr>
<td>LTRIM</td>
<td>247</td>
</tr>
<tr>
<td>MAKE_DATE_TIME</td>
<td>248</td>
</tr>
<tr>
<td>MD5</td>
<td>249</td>
</tr>
<tr>
<td>METAPHONE</td>
<td>250</td>
</tr>
<tr>
<td>MOD</td>
<td>253</td>
</tr>
<tr>
<td>NPER</td>
<td>254</td>
</tr>
<tr>
<td>PMT</td>
<td>255</td>
</tr>
<tr>
<td>POWER</td>
<td>256</td>
</tr>
<tr>
<td>PV</td>
<td>257</td>
</tr>
<tr>
<td>RAND</td>
<td>258</td>
</tr>
<tr>
<td>RATE</td>
<td>258</td>
</tr>
<tr>
<td>REG_EXTRACT</td>
<td>259</td>
</tr>
<tr>
<td>REG_MATCH</td>
<td>261</td>
</tr>
<tr>
<td>REG_REPLACE</td>
<td>262</td>
</tr>
<tr>
<td>REPLACECHR</td>
<td>263</td>
</tr>
<tr>
<td>REPLACESTR</td>
<td>265</td>
</tr>
<tr>
<td>REVERSE</td>
<td>267</td>
</tr>
<tr>
<td>ROUND (Dates)</td>
<td>268</td>
</tr>
<tr>
<td>ROUND (Numbers)</td>
<td>270</td>
</tr>
<tr>
<td>RPAD</td>
<td>271</td>
</tr>
<tr>
<td>RTRIM</td>
<td>272</td>
</tr>
<tr>
<td>SET_DATE_PART</td>
<td>274</td>
</tr>
<tr>
<td>SIGN</td>
<td>275</td>
</tr>
<tr>
<td>SIN</td>
<td>276</td>
</tr>
<tr>
<td>SINH</td>
<td>277</td>
</tr>
<tr>
<td>SOUNDEX</td>
<td>277</td>
</tr>
<tr>
<td>SQRT</td>
<td>279</td>
</tr>
<tr>
<td>SUBSTR</td>
<td>279</td>
</tr>
<tr>
<td>SYSTIMESTAMP</td>
<td>281</td>
</tr>
<tr>
<td>TAN</td>
<td>282</td>
</tr>
<tr>
<td>TANH</td>
<td>282</td>
</tr>
<tr>
<td>TO_BIGINT</td>
<td>283</td>
</tr>
<tr>
<td>TO_CHAR (Dates)</td>
<td>284</td>
</tr>
</tbody>
</table>
TO_CHAR (Numbers) ..................................................... 287
TO_DATE ...................................................................... 288
TO_DECIMAL .......................................................... 290
TO_FLOAT ............................................................ 290
TO_INTEGER .......................................................... 291
TRUNC (Dates) ......................................................... 292
TRUNC (Numbers) ....................................................... 294
UPPER ...................................................................... 295

Appendix C: Operators .................................................. 296
Operator Precedence ..................................................... 296
Arithmetic Operators ...................................................... 297
String Operators ......................................................... 298
  Nulls ..................................................................... 298
  Example of String Operators ........................................ 298
Comparison Operators ..................................................... 298
Logical Operators ........................................................ 299
  Nulls ..................................................................... 299

Appendix D: Constants .................................................... 300
FALSE ............................................................... 300
  Example ................................................................ 300
NULL................................................................ 300
  Working with Null Values in Boolean Expressions. .............. 301
  Null Values in Filter Conditions. .................................. 301
  Nulls with Operators ................................................ 301
TRUE................................................................ 301
  Example ................................................................ 301

Appendix E: Dates ......................................................... 302
Dates Overview ......................................................... 302
  Date/Time Datatype .................................................. 302
  Julian Day, Modified Julian Day, and the Gregorian Calendar. . 303
  Dates in the Year 2000 ................................................. 303
  Dates in Databases ................................................... 304
  Dates in Flat Files ..................................................... 305
  Default Date Format ................................................ 305
Date Format Strings ...................................................... 305
TO_CHAR Format Strings ................................................ 306
  Examples .................................................................. 308
TO_DATE and IS_DATE Format Strings .................................. 309
  Requirements .......................................................... 311
  Example ................................................................... 311

Table of Contents xi
Preface

Welcome to Informatica Cloud, Informatica’s hosted data services. Informatica Cloud uses functionality from PowerCenter and Informatica Data Quality to provide easy-to-use, web-based data services.

The Informatica Cloud User Guide explains how organization administrators and business users can use Informatica Cloud Services to perform tasks.

Informatica Resources

Informatica Documentation

The Informatica Documentation team takes every effort to create accurate, usable documentation. If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com. We will use your feedback to improve our documentation. Let us know if we can contact you regarding your comments.

The Documentation team updates documentation as needed. To get the latest documentation for your product, navigate to Product Documentation from http://mysupport.informatica.com.

Informatica Web Site

You can access the Informatica corporate web site at http://www.informatica.com. The site contains information about Informatica, its background, upcoming events, and sales offices. You will also find product and partner information. The services area of the site includes important information about technical support, training and education, and implementation services.

Informatica Cloud Web Site

You can access the Informatica Cloud web site at http://www.informaticacloud.com. This site contains information about Informatica Cloud editions and services. It also provides information about partners, customers, and upcoming events.

Informatica Cloud Community

You can access the Informatica Cloud Community at https://community.informatica.com/community/products/informatica_cloud.

Use the Community to discuss and resolve technical issues in Informatica Cloud. You can also find technical tips, documentation updates, and answers to frequently asked questions.
Informatica Cloud Trust Site

You can access the Informatica Cloud trust site at http://trust.informaticacloud.com. This site provides real-time information about Informatica Cloud system availability, current and historical data about system performance, and details about Informatica Cloud security policies.

Informatica Global Customer Support

To access Informatica Global Customer Support:

- Click the Support link in the Informatica Cloud application.
- Use the WebSupport Service. WebSupport requires a user name and password. You can request a user name and password at http://my.informatica.com.
- Call 1-888-345-4639.
Getting Started

This chapter includes the following topics:

- Understanding Informatica Cloud, 1
- Organization Administrator and User Accounts, 3
- Understanding the Informatica Cloud Tabs, 4
- Quick Setup, 5
- Help, Support, and Subscribe, 5
- Configuring User Preferences, 6

Understanding Informatica Cloud

Informatica Cloud is an on-demand subscription service that provides data services. When you subscribe to Informatica Cloud, you use a web browser to connect to the Informatica Cloud application. The Informatica Cloud applications runs at a hosting facility.

Informatica Cloud Components

Informatica Cloud includes the following components:

- Informatica Cloud application
  A browser-based application that runs at the Informatica Cloud hosting facility. It allows you to configure connections, create users, and create, run, schedule, and monitor tasks.

- Informatica Cloud hosting facility
  A facility where the Informatica Cloud application runs. The Informatica Cloud hosting facility stores all task and organization information. Informatica Cloud does not store or stage source or target data.

- Informatica Cloud Services
  Services you can use to perform tasks, such as data assessment, data loading, data replication, and data synchronization.

- Informatica Cloud Secure Agent
  A component of Informatica Cloud installed on a local machine that runs all tasks and provides firewall access between the hosting facility and your organization. When the Secure Agent runs a task, it connects to the Informatica Cloud hosting facility to access task information, connects directly and securely to sources and targets, transfers data between sources and targets, and performs any additional task requirements.
Informatica Cloud Services

Informatica Cloud provides the following services:

- **Data Assessment service**
  Use to measure the quality of data in Salesforce. You configure and run data assessment tasks to measure field completeness, field conformance, record duplication, and address validity for each Salesforce object in a Salesforce account.

- **Data Loader service**
  Use to read data from and write data to Salesforce, databases, and flat files. For example, you can use the Data Loader service to load data from a flat file to Salesforce. You might also use the Data Loader service to synchronize Salesforce data with a target.

- **Data Synchronization service**
  Use to read data from sources and write data to targets. The Data Synchronization service contains all Data Loader service functionality. It also contains additional functionality, such as using multiple source objects, custom source objects, multiple tasks in a schedule, and additional source and target combinations.

- **Data Replication service**
  Use to replicate data from a Salesforce source to a target. You might replicate data to archive the data, perform offline reporting, or consolidate and manage data.

- **PowerCenter service**
  Use to import a PowerCenter workflow and run it as an Informatica Cloud PowerCenter task.

Informatica Cloud Subscriptions

Informatica offers several different subscription options to access Informatica Cloud. Depending on the subscription associated with your organization, you might not have access to the full range of Informatica Cloud functionality.

For more information about subscription options, trial users can click the Subscribe link that displays on all Informatica Cloud tabs. Users with an existing subscription can read about subscription options and register for different subscriptions at the following URL: [http://www.informaticacloud.com/products/editions.html](http://www.informaticacloud.com/products/editions.html).

Informatica Cloud Security

Informatica Cloud uses 128-bit Secure Sockets Layer (SSL) technology to protect data. It uses authentication and encryption to ensure that data is secure and available only to users within the organization. When you log in to Informatica Cloud, https precedes the URL in the address field of the web browser to indicate that the connection is secure.
Minimum System Requirements

The following table shows the minimum system requirements for components required to run each of the Informatica Cloud Services:

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser</td>
<td>- Mozilla Firefox 3.6 or Internet Explorer 6.x or 7.0</td>
</tr>
<tr>
<td></td>
<td>- Windows 2000, 2003, or XP operating system</td>
</tr>
<tr>
<td>Secure Agent</td>
<td>- Windows or Linux operating system (32-bit or 64-bit)</td>
</tr>
<tr>
<td></td>
<td>- 500MB disk space</td>
</tr>
<tr>
<td></td>
<td>- Internet access to the web server that runs the Informatica Cloud application</td>
</tr>
<tr>
<td></td>
<td>- Network connectivity to source and target systems</td>
</tr>
</tbody>
</table>

Organization Administrator and User Accounts

The organization administrator is the Informatica Cloud user responsible for managing the organization. The organization administrator is configured with the Admin role and full access to the organization. If an organization has multiple users with the Admin role and full access to the organization, the organization administrator is the lead administrator.

By default, the user who creates the organization is the organization administrator. When a user signs up for an organization with Informatica Cloud, Informatica Cloud assigns the user the Admin role and full access to the organization.

The organization administrator can create a user account for each non-administrator user who needs to access the organization. The organization administrator can configure roles for each user to determine the tasks that the user can perform. If the organization has the Fine-Grained Security license, the organization administrator can configure user groups and object-level permissions. If the organization has the Org Hierarchy license, the organization administrator can create sub-organizations and manage all organizations in the organization hierarchy.

**Related Topics:**
- “Users” on page 21

User Login

Before you log in, verify that the browser meets the minimum system requirements. The first-time login process differs based on the following user types:

- Organization administrator
- Non-administrator user

The login process is the same for any user after the first login.

After you log in, the Informatica Cloud Messages page may appear. The Informatica Cloud Messages page shows maintenance notices, such as when the Informatica Cloud application will be unavailable because of an upgrade. From this page, you can click Continue to begin using Informatica Cloud.
Logging In as the Organization Administrator

To create an Informatica Cloud account and register your organization with Informatica Cloud, click the Don’t Have an Account link on the Login page.

When you register an organization, Informatica Cloud creates an organization administrator account using your email address as the user name and generates a temporary password. Informatica Cloud sends the user name, password, and Informatica Cloud URL to your email address. To log in to Informatica Cloud for the first time, click the Login with Temporary Password link in the email, or enter the user name and password provided in the email.

After you log in as the organization administrator for the first time, Informatica Cloud requires you to change the password. After you change the password, you can start using Informatica Cloud.

Logging In as a Non-Administrator User

Before you log in to Informatica Cloud, the organization administrator must create your user account and assign you a user name and password. To log in to Informatica Cloud, enter your user name and password in the Login page. Your user name is an email address.

Account Passwords

Depending on your user profile configuration, you may need to change your password the first time you log in. If you forget your password, click the "Forgot your password" link on the Login page to reset your password. When you reset your password, you must enter your user name. Informatica Cloud sends the new password to your email address.

Account Lockouts

You may get locked out of your account if you exceed the maximum number of failed login attempts. If you are locked out, contact the organization administrator to reset your password. If you are the organization administrator, contact Informatica Global Customer Support to reset your password.

License Expiration

An organization can have several licenses. When a license expires, you cannot access the service or features associated with the license. If all licenses for the organization expire, you cannot log in to Informatica Cloud.

The organization administrator can review the expiration date for licenses. To extend a license, contact Informatica Global Customer Support.

Understanding the Informatica Cloud Tabs

When you log in to Informatica Cloud, the Main page usually appears. At times, the Informatica Cloud Messages page might appear to inform you about upcoming maintenance.

Depending on your role, the Main page includes the following tabs:

- **Home**
  Monitor jobs to view details about tasks that are running or complete. You can also view scorecards for all data assessment tasks that ran to determine the quality of the Salesforce data.

- **Data Services**
  Access Informatica Cloud Services.
Quick Setup

Use the Quick Setup page to quickly set up Informatica Cloud. The Quick Setup page lists the following high-level steps required to set up each service:

1. Select one of the Informatica Cloud Services.
2. Install the Secure Agent.
3. Create a task.
4. Run the task.

This guide discusses each of these concepts in detail.

Help, Support, and Subscribe

Informatica Cloud provides quick access to online help and a Support page. Trial users can also access a Subscription page.

Online Help

You can click the Help link on any page to view documentation about the task you are performing. Through the online help, you can access all Informatica Cloud documentation. The online help also provides links to the Informatica Cloud Community, Informatica Cloud YouTube channel, and Informatica Cloud Global Customer Support contact information.

Support Page

You can click the Support link on any page to view the Support page. The Support page provides links to the Informatica Cloud online help, the Informatica Cloud Community, and the Informatica Cloud YouTube channel.

Users with a paid license or those within the 30-day trial period can submit a support request directly to Informatica Global Customer Support. When you submit a support request, you indicate the severity and priority of the issue and enter a full description of the support request.

Informatica Cloud Subscription Options Page

Trial users can click the Subscribe link on any Informatica Cloud tab to view the Informatica Cloud Subscription Options page. This page provides information about the Informatica Cloud subscription options.
On the Informatica Cloud Subscription Options page, trial users can subscribe to Informatica Cloud Express and contact us about other subscription options.

Users with an existing subscription can read about subscription options and register for different subscriptions at the following URL: http://www.informaticacloud.com/products/editions.html.

**Subscribing to Informatica Cloud Express**

Trial users can subscribe to Informatica Cloud Express through Informatica Cloud. After you subscribe to Informatica Cloud Express, you can verify the number of rows processed by your organization for the current month at any time. On the View Organization page, check the value for the Monthly Job Row Count property.

To subscribe to Informatica Cloud Express:

1. On any Informatica Cloud tab, click the **Subscribe** link.
   The Informatica Cloud Subscription Options page displays.
2. Under Informatica Cloud Express, click **Buy Now**.
   The Buy Informatica Express Now page displays Informatica Cloud Express pricing information and frequently asked questions.
3. To subscribe to Informatica Cloud Express, enter required billing information, read and agree to the subscription agreement, and click **Buy Now**.

**Configuring User Preferences**

User preferences are the details of your Informatica Cloud user account.

You can perform the following tasks on user preferences:

- View user preferences.
- Edit user preferences.

**Viewing User Preferences**

View user preferences to see details about your user account, including your name, default views, and the time zone associated with your user account.

To view user preferences:

- Click your user name link at the top of the browser window.

Your user preferences appear on the View Preferences page.

**Editing User Preferences**

Edit user preferences to configure your user profile. User preferences include contact information, password, default views, and the time zone associated with your user account.

1. Click your user name link at the top of the browser window, and then click **Edit**.
2. On the Update Preferences page, configure the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>Informatica Cloud user name. You cannot update your user name.</td>
</tr>
<tr>
<td>Salesforce User Name</td>
<td>User name for your Salesforce account.</td>
</tr>
<tr>
<td>Confirm Salesforce User Name</td>
<td>Confirmation of your Salesforce account.</td>
</tr>
<tr>
<td>Password</td>
<td>Informatica Cloud password.</td>
</tr>
<tr>
<td>Confirm Password</td>
<td>Confirmation of your Informatica Cloud password.</td>
</tr>
<tr>
<td>First Name</td>
<td>First name.</td>
</tr>
<tr>
<td>Last Name</td>
<td>Last name.</td>
</tr>
<tr>
<td>Job Title</td>
<td>Job title.</td>
</tr>
<tr>
<td>Phone Number</td>
<td>10-digit telephone number. For example, (123) 456-7890, 123-456-7890, or 123.456.7890.</td>
</tr>
<tr>
<td>Emails</td>
<td>Email addresses that receive notification when the password is reset for this user account. Separate multiple email addresses with commas. If this field is blank, Informatica Cloud sends password reset information to the email address that you specify as the user name for the Informatica Cloud account.</td>
</tr>
<tr>
<td>Time Zone</td>
<td>Time zone where you are located. When you log in, all times reflect the time zone that you specify. For example, the dashboard and schedules display the local time in this time zone. Default time zone is Pacific Daylight Time, Los Angeles.</td>
</tr>
</tbody>
</table>
| Ignore License Expiration Messages | Allows you to stop the display of license expiration messages:  
  - False. Shows all license expiration messages.  
  - True. Does not display license expiration messages. |
| Data Synchronization Tasks View | Default view for data synchronization tasks. You can select any available view. |
| Data Replication Tasks View  | Default view for data replication tasks. You can select any available view.  |
| Data Assessment Tasks View   | Default view for data assessment tasks. You can select any available view.   |
| PowerCenter Tasks View       | Default view for PowerCenter tasks. You can select any available view.       |
| Connections View             | Default view for connections. You can select any available view.             |
| Task Flows View              | Default view for task flows. You can select any available view.              |
CHAPTER 2

Performing Administrative Tasks

This chapter includes the following topics:

- Organization Profile, 8
- Organization Security Overview, 14
- Roles, 16
- User Groups, 17
- Object-Level Permissions, 19
- Users, 21
- Migrating Objects, 23
- Viewing the Audit Log, 25
- Billing Information, 25
- Organization Hierarchy, 27

Organization Profile

The organization administrator can view and configure the organization profile:

- View organization properties.
- Configure organization properties and details.
- View license information.

Viewing Organization Properties

You can view the properties of the organization, including the organization ID, organization type, maximum daily jobs, and monthly job row count.

- Click Administration > Organization.
The following organization properties appear on the View Organization page:

<table>
<thead>
<tr>
<th>Organization Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization ID</td>
<td>Uniquely identifies the organization in the system. Informatica Cloud generates the organization ID.</td>
</tr>
<tr>
<td>Parent Organization ID</td>
<td>The ID of the parent organization. Organization hierarchies only.</td>
</tr>
<tr>
<td>Dev Org</td>
<td>Indicates the organization is a development organization. If it is not a development organization, it is a production organization.</td>
</tr>
<tr>
<td>Publishing Org</td>
<td>Indicates the organization can publish plug-ins and custom sources to other organizations.</td>
</tr>
<tr>
<td>Schedule Offset</td>
<td>The small amount of time added to the start time of each schedule. A schedule runs based on the configured start time and schedule offset. Schedule offset does not affect the start time of manually started tasks or task flows.</td>
</tr>
<tr>
<td>Maximum Daily Jobs*</td>
<td>The maximum number of jobs the organization can run every day. This includes both successful and failed jobs. Default is 25 for trial organizations.</td>
</tr>
<tr>
<td>Maximum Daily Job Rows*</td>
<td>The maximum number of rows the organization can process every day. Default is 10,000 for trial organizations.</td>
</tr>
<tr>
<td>Maximum Monthly Job Rows*</td>
<td>The maximum number of rows the organization can process every month. Default is 300,000 for trial organizations.</td>
</tr>
<tr>
<td>Daily Job Count</td>
<td>The number of jobs run for the current day.</td>
</tr>
<tr>
<td>Monthly Job Count</td>
<td>The number of jobs run for the current month.</td>
</tr>
<tr>
<td>Daily Job Row Count</td>
<td>The number of rows processed for the current day.</td>
</tr>
<tr>
<td>Monthly Job Row Count</td>
<td>The number of rows processed for the current month.</td>
</tr>
<tr>
<td>Maximum Sub-Organizations**</td>
<td>The maximum number of sub-organizations allowed for the organization. Organization hierarchies only.</td>
</tr>
<tr>
<td>Maximum Log Entries</td>
<td>Maximum number of entries that you want to appear in the activity log. Click Clear Log to clear the activity log.</td>
</tr>
<tr>
<td>Created On</td>
<td>Date and time the organization was created.</td>
</tr>
<tr>
<td>Updated On</td>
<td>Date and time the organization was updated.</td>
</tr>
<tr>
<td>Created By</td>
<td>User who created the organization.</td>
</tr>
<tr>
<td>Updated by</td>
<td>User who updated the organization.</td>
</tr>
</tbody>
</table>

* To increase the number of jobs the organization can run or the number of rows it can process, you can subscribe to Informatica Cloud. Use the Subscribe link or contact Informatica Global Customer Support.

** To increase the number of sub-organizations allowed in the organization hierarchy, contact Informatica Global Customer Support.
Schedule Offset

The schedule offset is a small amount of time that is added to schedule start times. An organization has a single schedule offset that is applied to all schedules. The schedule offset does not affect the start time of manually started tasks or task flows. The schedule offset helps prevent server overload at standard schedule start times.

Though not displayed, the schedule offset for your organization is also added to the time range configured for all schedules. This ensures that scheduled tasks run as often as expected. For example, you configure a schedule to run every hour from 8:00 a.m. to 12:00 p.m and the schedule offset for your organization is 15 seconds. Your schedule runs at 8:00:15, 9:00:15, 10:00:15, 11:00:15, and 12:00:15.

To view the schedule offset for your organization:

- Click Administration > Organization.
  
  The Schedule Offset property indicates the amount of time that is added to the start time of each schedule.

Maximum Daily Jobs

An organization can run a certain number of jobs each day. When your organization reaches the daily limit, you cannot run additional jobs until the next day. By default, an organization can run up to 25 jobs each day.

To verify the number of jobs your organization can run each day and the number of jobs that have run on the current day:

- Click Administration > Organization.
  
  On the View Organization page, the Maximum Daily Jobs property displays the number of jobs your organization can run each day. The Daily Job Count property displays the number of jobs that have already run that day.
  
  To increase the number of daily jobs allowed for your organization, contact Informatica Global Customer Support.

Configuring Organization Properties and Details

You can edit organization properties, such as maximum log entries. You can also edit organization details, default email notification options, and authentication options.

Understanding Email Notification

You can configure Informatica Cloud to send email notification when tasks complete successfully, complete with errors, or fail. You can configure email notification at an organizational level. You can also configure email notification for an individual task or task flow. When you specify email notification in a task or task flow, Informatica Cloud sends email to the addresses in the task or task flow instead of the addresses configured for the organization.

The email notification includes the status and results of the task. It also includes the start time and end time for the task. Informatica Cloud uses the time zone of the user who created the task. If the user profile no longer exists, Informatica Cloud uses UTC time.

Understanding Password Complexity

To configure password complexity for all users in the organization, configure the Minimum Character Mix organization property.
Passwords can contain a mix of the following character sets:

- Lowercase letters
- Capital letters
- Numbers
- Special characters

When you configure password complexity, select one of the following options:

<table>
<thead>
<tr>
<th>Minimum Character Mix Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Contains at least one of the character sets.</td>
</tr>
<tr>
<td>2</td>
<td>Contains at least two of the character sets.</td>
</tr>
<tr>
<td>3</td>
<td>Contains at least three of the character sets.</td>
</tr>
<tr>
<td>4</td>
<td>Contains all four of the character sets.</td>
</tr>
</tbody>
</table>

Understanding Two-Factor Authentication

Two-factor authentication uses trusted IP address ranges in addition to account passwords to enhance Informatica Cloud security. Configure the Two-Factor Authentication organization property when you want to increase security for your organization.

When you configure two-factor authentication, you enter one or more trusted IP address ranges. A user with a valid login must have an IP address within the trusted IP address ranges to connect to your organization.

Informatica Cloud supports IP address formats in IP version 4 (IPv4) and version 6 (IPv6).

**Note:** Entering an inappropriate IP address range can prevent all users from accessing your organization. Contact your network administrator for valid IP address ranges.

Steps to Configure Organization Properties and Details

To configure organization properties and details:

1. Click **Administration > Organization**, and then click **Edit**.
2. On the Edit Organization page, update organization properties and details.

The following table describes organization properties:

<table>
<thead>
<tr>
<th>Organization Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization ID</td>
<td>Uniquely identifies the organization in the system. Informatica Cloud generates the organization ID.</td>
</tr>
<tr>
<td>Maximum Log Entries</td>
<td>Maximum number of entries that you want to list in the activity log. The maximum applies to all users in the organization that view the activity log. Enter an integer between 1 and 1000. Default is 100. You can clear the number of log entries on the View Organization page.</td>
</tr>
</tbody>
</table>
The following table describes organization details:

<table>
<thead>
<tr>
<th>Organization Detail</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization Name</td>
<td>Name of organization.</td>
</tr>
<tr>
<td>Address1</td>
<td>Address of organization.</td>
</tr>
<tr>
<td>Address2</td>
<td>Additional address information for the organization.</td>
</tr>
<tr>
<td>Address3</td>
<td>Additional address information for the organization.</td>
</tr>
<tr>
<td>City</td>
<td>City where the organization is based.</td>
</tr>
<tr>
<td>State</td>
<td>State where the organization is based.</td>
</tr>
<tr>
<td>Postal Code</td>
<td>Postal code of the area where the organization is based.</td>
</tr>
<tr>
<td>Country</td>
<td>Country where the organization is based.</td>
</tr>
<tr>
<td>Employees</td>
<td>Number of employees in the organization.</td>
</tr>
<tr>
<td>Do You Use Salesforce</td>
<td>Indicates whether your organization uses Salesforce.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the organization. Maximum length is 255 characters.</td>
</tr>
</tbody>
</table>

The following table describes default email notification options:

<table>
<thead>
<tr>
<th>Email Notification Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Email Notification</td>
<td>List of comma-separated email addresses that receive email notification when a task fails to complete. For example, <a href="mailto:admin@acme.com">admin@acme.com</a>, <a href="mailto:sa@acme.com">sa@acme.com</a>.</td>
</tr>
<tr>
<td>Warning Email Notification</td>
<td>List of comma-separated email addresses that receive email notification when a task completes with errors.</td>
</tr>
<tr>
<td>Success Email Notification</td>
<td>List of comma-separated email addresses that receive email notification when a task completes without errors.</td>
</tr>
</tbody>
</table>

The following table describes authentication options:

<table>
<thead>
<tr>
<th>Authentication Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Character Mix</td>
<td>Passwords can contain a mix of the following character sets: lowercase letters, capital letters, numbers, and special characters. Select the mix of characters that each password must contain: - 1. Contains at least one of the character sets. - 2. Contains at least two of the character sets. - 3. Contains at least three of the character sets. - 4. Contains all four character sets.</td>
</tr>
</tbody>
</table>
## Authentication Options

<table>
<thead>
<tr>
<th>Authentication Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default is 1.</td>
<td></td>
</tr>
<tr>
<td>Password Reuse Allowed</td>
<td>Number of days until a previous password can be used again. Default is Always.</td>
</tr>
<tr>
<td>Password Expires</td>
<td>Number of days until the password expires. Default is Never.</td>
</tr>
<tr>
<td>Two-Factor Authentication</td>
<td>A security option that allows the configuration of trusted IP address ranges as an addition to account password authentication: - Disabled. Informatica Cloud requires account passwords for access to the organization. - Enabled. Informatica Cloud requires account passwords for access to the organization, and the user must login from within configured IP address ranges.</td>
</tr>
<tr>
<td>Trusted IP Ranges</td>
<td>Enabled when Two-Factor Authentication is selected. Enter one or more trusted IP address ranges. You can enter IP address ranges that use IP format version 4 (IPv4) or version 6 (IPv6). Use the Add icon to add another range. Use the Remove icon to remove a range.</td>
</tr>
</tbody>
</table>

3. Click **OK**.

## Understanding License Types

When you create an Informatica Cloud account, Informatica Cloud assigns the organization a license type for each data service.

Informatica Cloud uses the following types of licenses:

- **Trial**
  You can use the data service free of charge for a 30-day period. At the end of the trial period, you can subscribe to the service. A trial subscription may provide limited access to the features of the data service.

- **Subscription**
  You can use the data service for the duration of the contract period. Near the end of the contract period, Informatica Cloud indicates that the contract is about to expire. Renew the contract to continue using the service.

- **Free subscription**
  You can use the Data Loader service free of charge. A free subscription may provide limited access to the features of the Data Loader service.

## Viewing License Information

You can view the license information for your organization, as well as the support level and expiration date for each license.

- Click **Administration > Organization**.
On the View Organization page, the Service Licenses table displays the following information:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>License name and the Informatica Cloud service or functionality allowed by the license:</td>
</tr>
<tr>
<td></td>
<td>- DSS. Data synchronization service.</td>
</tr>
<tr>
<td></td>
<td>- DRS. Data replication service.</td>
</tr>
<tr>
<td></td>
<td>- DQA. Data assessment service.</td>
</tr>
<tr>
<td></td>
<td>- PCS. PowerCenter service.</td>
</tr>
<tr>
<td></td>
<td>- Plug-in. Plug-ins.</td>
</tr>
<tr>
<td></td>
<td>- Custom source. Custom sources.</td>
</tr>
<tr>
<td></td>
<td>- DNB360. DNB360.</td>
</tr>
<tr>
<td></td>
<td>- Migration. Allows migrating objects.</td>
</tr>
<tr>
<td></td>
<td>- Org Hierarchy. Allows working with sub-organizations and the Informatica Cloud Administrator.</td>
</tr>
<tr>
<td></td>
<td>- Fine-Grained Security. Allows working with user groups and object-level permissions.</td>
</tr>
<tr>
<td>License Type</td>
<td>Type of Informatica Cloud license.</td>
</tr>
<tr>
<td>Support Level</td>
<td>Level of customer support for the license. Values are Online, Standard, and Premium.</td>
</tr>
<tr>
<td>Expires On</td>
<td>Date when the license expires. If all licenses expire, the account is disabled.</td>
</tr>
<tr>
<td></td>
<td>To extend a license, contact Informatica Global Customer Support.</td>
</tr>
<tr>
<td>Enabled</td>
<td>Indicates if the data service or feature associated with the license is enabled.</td>
</tr>
</tbody>
</table>

## Organization Security Overview

You can use a combination of roles, user groups, and object-level permissions to secure the objects and data for your organization.

- **Role.** A set of privileges defined by Informatica Cloud. An organization administrator must assign at least one role to each user.

- **User group.** A set of permissions for object types in the organization. An organization administrator can create user groups with logical sets of permissions and assign user groups to users. This allows you to restrict the access provided by the role assigned to a user. You can assign one or more user groups for a user, but best practice is to assign a maximum of one group to a user. Not all users require a user group.

- **Object-level permissions.** Permissions that you can configure for individual objects in the organization. Configure permissions for objects when you need additional or custom security for specific objects. With object-level permissions, you can expand or restrict user group permissions for an object.

The role assigned to a user defines the general tasks that the user can perform in the organization. The user group assigned to a user defines the objects that the user can work with and the tasks that the user can perform with those objects. Object-level permissions allow you to add or remove individual objects from the user group domain.

In addition to these security measures, you can increase the requirements for accessing the organization with the Two-Factor Authentication and Minimum Character Mix organization properties. You can also limit the ability to preview data with the Allow Data Preview user security option.
Security Examples and Tips

**Note:** In any scenario, best practice is to assign the Admin role to one or two trusted users without assigning them to user groups that might restrict their access. These users can act as alternative organization administrators and can help troubleshoot access control and other issues that might occur when increasing organization security.

You want your development team to create tasks and task flows. To do this, the development team needs to view sample data in development, but you want to restrict access to production data. You also have a reporting team that needs to run tasks, but does not have the technical knowledge to configure tasks safely.

For these two teams, create two separate user groups. For the development team, create a Developer user group and control the user group as follows:

1. Configure the Developer group with full permissions for tasks and related objects, but only read permission for connections.
2. Assign the Designer role and the Developer group to the users in your development team. When configuring users, select the Allow Preview Data option to help with task development.
3. If possible, create development connections to sample data. If you have both development and production connections, configure production connections so the Developer group does not have read permission. This prevents the Developer group from using production connections in tasks.
4. After testing is complete and tasks are ready to move into production, have an organization administrator or other qualified user configure the tasks to use production connections.
5. Edit the Developer user group to remove the permission to run tasks. If development is complete for a task type, you can also remove permission to read or update the task type and related objects as well. By removing the read permission, you prevent the Developer group from accessing any information about production tasks.

For the reporting team, create a Reporting user group, and control the group as follows:

1. Configure the Reporting group with permission to read and run tasks and task flows, and permission to read, create, and update schedules. Do not allow the group to edit or delete objects in the organization, or to change permissions for objects in the organization.
2. Assign the Designer role and the Reporting group to the users in your reporting team.

You want a site operator who can assign roles and user groups and configure access control, but cannot create, edit, or run tasks.

1. Assign the Admin role to the site operator user. With the Admin role, the site operator can configure users and user groups and manage the organization. But the Admin role also allows full access to tasks, connections, and schedules. To limit this access, assign the site operator to a restrictive Site Operator user group.
2. Configure the Site Operator user group as follows:
   - Assign permissions to read objects and to configure permissions for all object types. If the Site Operator group needs to download and install the Secure Agent, also grant the group create permission for Secure Agents.
   - Restrict all other permissions. By revoking the create, update, delete, or run permissions on objects, you limit the Site Operator group to performing administrative tasks.

Rules and Guidelines for Security

Use the following rules and guidelines for roles, user groups, and object-level permissions:

- Assign each user a single role.
Though you can assign more than one user group to a user, you can more easily manage the permissions assigned to a user by assigning the user to a single group.

Not all users need to be assigned to a user group.

To edit an object, the user also needs read permission for all objects used within the object. For example, when you assign a user the read and update permission for data synchronization tasks, verify that the user also has read permission for the connections, custom sources, plug-ins, and schedules to be used in those tasks.

When a user edits a task, objects without read permission do not display. To avoid unexpected results, the user should cancel all changes and avoid editing the task until you grant the appropriate read permissions.

When configuring a task flow, a user needs run permission on tasks to be added to the task flow.

To edit a task flow, a user needs run permission for all tasks in the task flow. Without run permission on all tasks, the user cannot save changes to the task flow.

To monitor jobs or to stop a running job, a user needs run permission for the task or task flow.

To migrate objects, the user account in both the source and target organizations should have the Admin role with full access to the repository.

When adding a new license to an organization, Informatica Cloud grants read, create, update, run, and delete permissions for the related functionality to all existing user groups. If you want to restrict new functionality to certain users, review user group permissions after receiving a new license.

For more information about permission types, see “Configuring Object-Level Permissions” on page 19.

Roles

A role is a set of privileges that allows a user to perform tasks in the organization. Assign each user at least one role. Best practice is to assign only one role to each user.

Roles determine the functionality and Informatica Cloud tabs that are available to a user. For example, to access the Administration tab and perform any tasks on the Administration tab, you need the Admin role.

Informatica Cloud provides the following roles:

- **Admin.** Has access to all Informatica Cloud functionality and can perform any task in the organization.
- **Designer.** Has limited access to Informatica Cloud functionality. Can configure and test tasks, configure all related objects, and monitor jobs. Cannot perform administrative tasks for the organization.

In addition to roles, user groups and object-level permissions can determine the ability of a user to view and perform tasks on an object.

The following table describes the privileges for each role:

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Admin Role</th>
<th>Designer Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access to all available functionality on the Home tab.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Access to all available functionality on the Data Services tab.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Access to all available functionality on the Configuration tab.</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Access to all available functionality on the Administration tab.</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
User Groups

A user group is a set of permissions that allows a user to perform tasks on objects in the organization. To work with user groups, your organization must have the Fine-Grained Security license.

The User Groups page displays a list of all user groups in the organization.

You can perform the following tasks for a user group:

- Configure a user group.
- Edit a user group.
- View user group details.
- Delete a user group.

For information about obtaining the Fine-Grained Security license, contact Informatica Global Customer Support.

Configuring a User Group

The organization administrator can create and configure user groups for the organization. When you configure a user group, you configure the name, description, and permissions for the group. You can configure permissions for the following objects:

- Secure Agents
- Connections
- Custom sources
- Plug-ins
- Schedules
- Task flows
- Data assessment tasks
- Data loader tasks
- Data replication tasks
- Data synchronization tasks
- PowerCenter tasks
The following table describes the types of permissions that you can configure:

<table>
<thead>
<tr>
<th>Permission</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Allows you to view objects and to view details about each object. Also allows you to use a connection or schedule in a task.</td>
</tr>
<tr>
<td>Create</td>
<td>Allows you to create objects. Also allows you to download and install the Secure Agent.</td>
</tr>
<tr>
<td>Update</td>
<td>Allows you to edit objects. Requires read permission, which is automatically granted.</td>
</tr>
<tr>
<td>Delete</td>
<td>Allows you to delete objects. Requires read permission, which is automatically granted.</td>
</tr>
<tr>
<td>Run</td>
<td>Allows you to run objects. Allows you to monitor jobs and stop jobs in the activity monitor. Also allows you to add tasks to a task flow if you have update permission on the task flow. Requires read permission, which is automatically granted.</td>
</tr>
<tr>
<td>Change Permission</td>
<td>Allows you to configure permissions for individual objects of the selected type. Requires read permission, which is automatically granted.</td>
</tr>
</tbody>
</table>

**Note:** These permissions control what is allowed in Informatica Cloud. They do not control operating system permissions, such as the ability to start, stop, or configure the Secure Agent on Windows or Linux.

To configure a user group:

1. To create a user group, click Administration > User Groups, and then click New.
   To edit a user group, click Administration > User Groups, click the user group you want to edit, and then click Edit.
2. On the New User Group page or Edit User Group page, configure the following details:

<table>
<thead>
<tr>
<th>User Group Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the user group. User group names are not case sensitive. The user group name must be unique within the organization. User group names can contain alphanumeric characters, spaces, and the following special characters: _ - +.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the user group.</td>
</tr>
</tbody>
</table>

3. Select the permissions that you want to grant to the user group.
   If you select a permission that requires Read permission, Read permission is automatically selected. For example, to delete a connection you must be able to view the connection, so when you select Delete permission, Read permission is also selected.
   Permissions that are not relevant for the object type are disabled.

4. Click OK.
Viewing User Group Details

You can view details about a user group, such as the name and permissions for the user group. When you view user group details, you can click Edit to update the details.

- Click Administration > User Groups, and then click the name of the user group.

User group details appear on the View User Group page.

Deleting a User Group

Delete a user group if you no longer need the user group. Remove all users from the user group before you delete the user group.

1. Click Administration > User Groups.
2. On the User Group page, click the Delete icon next to the name of the user group.

Object-Level Permissions

You can configure permissions for individual objects in the organization. To work with object-level permissions, your organization must have the Fine-Grained Security license.

By default, the user group assigned to a user defines the objects that the user can access. User group permissions are configured by object type.

When you configure permissions for an individual object, you can restrict the permissions the user group has for the object. You can also expand the permissions a user group has for the object.

For example, you have a Developer user group that allows all users in the group to configure, delete, and run all tasks. You also have a Scheduling user group that allows its users to configure schedules and task flows, but not tasks. When a task no longer requires editing by the Developer user group, you can secure the task and move it into production by editing task permissions. Configure the task permissions to remove all permissions for the Developer group. At the same time, you can grant the Scheduling group view and run permission.

For information about obtaining the Fine-Grained Security license, contact Informatica Global Customer Support.

Related Topics:
- “Roles” on page 16
- “User Groups” on page 17
- “Organization Security Overview” on page 14

Configuring Object-Level Permissions

An organization administrator can configure user group access to individual objects in the organization.

You can configure permissions for the following types of objects:

- Secure Agents
- Connections
- Custom sources
- Plug-ins
The following table describes the types of permissions that you can configure:

<table>
<thead>
<tr>
<th>Permission</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>Allows you to view objects and to view details about each object. Also allows you to use a connection or schedule in a task.</td>
</tr>
<tr>
<td>Create</td>
<td>Allows you to create objects. Also allows you to download and install the Secure Agent.</td>
</tr>
<tr>
<td>Update</td>
<td>Allows you to edit objects. Requires read permission, which is automatically granted.</td>
</tr>
<tr>
<td>Delete</td>
<td>Allows you to delete objects. Requires read permission, which is automatically granted.</td>
</tr>
<tr>
<td>Run</td>
<td>Allows you to run objects. Allows you to monitor jobs and stop jobs in the activity monitor. Also allows you to add tasks to a task flow if you have update permission on the task flow. Requires read permission, which is automatically granted.</td>
</tr>
<tr>
<td>Change Permission</td>
<td>Allows you to configure permissions for individual objects of the selected type. Requires read permission, which is automatically granted.</td>
</tr>
</tbody>
</table>

**Note:** These permissions control what is allowed in Informatica Cloud. They do not control operating system permissions, such as the ability to start, stop, or configure the Secure Agent on Windows or Linux.

To configure permissions for an object:

1. Navigate to the object that you want to configure. For example, to configure permissions for a schedule, click Configuration > Schedules to view the Schedules page.
2. Click the **Change Permission** icon for the object.
   The Change Permissions page displays the user groups in the organization.
   If custom permissions have not been configured for the object, the Default Permissions option is selected, and the Permissions table displays the permissions that each user group has for the object by default.
   If custom permissions have already been configured for the object, the Custom Permissions option is selected, and the Permissions table displays the custom permissions configured for each user group.
3. To change the permissions, click **Custom Permissions**.
   The Permissions table displays all relevant permissions for the object. Permissions that are not relevant are disabled. For example, since the object already exists, the Create permission is disabled.
4. Configure the permissions that you want to allow for each user group.
   To revert to the default permissions configured for all user groups, click Default Permissions.
5. Click **OK**.
Users

A user is an Informatica Cloud user account that allows secure access to an organization. A user can perform tasks in the organization based on the role and user group assigned to the user. Object-level permissions can also affect the tasks a user can perform.

The Users page displays a list of all Informatica Cloud users in the organization.

You can perform the following tasks for a user:

- Create a user.
- Edit a user.
- View user details.
- Delete a user.

**RELATED TOPICS:**

- “Roles” on page 16
- “User Groups” on page 17
- “Organization Security Overview” on page 14

Configuring a User

The organization administrator can configure users for the organization. When you configure a user, you configure the following information:

- **User properties.** Properties such as the user name, password, and maximum login retry attempts.
- **User details.** Information about the user, such as the name, job title, and phone number of the user.
- **User security.** Security options, such as the role, user group, and ability to preview data.

The role for a user defines the general tasks that the user can perform in the organization. The user group for a user defines the objects the user can work with and the specific tasks that the user can perform. For example, a user with the Admin role can perform any task in the organization. To limit the tasks that he can perform, you can assign him to a user group that can configure security and manage the organization, but cannot create or run tasks.

To configure a user:

1. To create a user, click Administration > Users, and then click New.
   To edit a user, click Administration > Users, click the user name in the list of users, and then click Edit.
2. On the New User page or Edit User page, configure the following user properties:

<table>
<thead>
<tr>
<th>User Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>Informatica Cloud user name that identifies the user within the organization. Used to log in to Informatica Cloud. The user name must be a valid email address and unique within Informatica Cloud. For example, <a href="mailto:john@abc.com">john@abc.com</a>. This field is read-only when you edit a user.</td>
</tr>
<tr>
<td>Salesforce User Name</td>
<td>Salesforce user name.</td>
</tr>
</tbody>
</table>
### User Property | Description
--- | ---
Confirm Salesforce User Name | Confirmation of Salesforce user name.
Password | Password for the user account. The password cannot be the same as the user name.
Confirm Password | Confirmation of the password.
Maximum Login Attempts | Maximum number of login attempts the user can make before the user is locked out. If locked out, the user must contact Informatica Global Customer Support to reset the account.
Change Password on Login | Determines if the user must reset the password after logging in. Select this option to require a user to create a new password after logging in.
Ignore License Expiration Messages | Allows you to stop the display of license expiration messages:
- False. Shows all license expiration messages.
- True. Does not display license expiration messages.

3. **Configure the following user details:**

### User Detail | Description
--- | ---
First Name | First name of the user.
Last Name | Last name of the user.
Job Title | Job title of the user.
Phone Number | 10-digit telephone number of the user. For example, (123) 456-7890, 123-456-7890, or 123.456.7890.
Emails | Email addresses that receive notification when the password is reset for this user account. Separate multiple email addresses with commas. If this field is blank, Informatica Cloud sends password reset information to the email address that you specify as the user name.
Time Zone | Time zone where the user is located. When the user logs in, all times reflect the time zone of the user. For example, schedules display the local time in the user time zone. Default time zone is Pacific Daylight Time, Los Angeles.
Description | Description of the user. Maximum length is 255 characters.

4. **Configure the following user security options:**

### User Security Option | Description
--- | ---
User Roles | Informatica Cloud role for the user. Select one of the following roles:
- **Admin.** Has access to all Informatica Cloud functionality and can perform any task in the organization.
- **Designer.** Has limited access to Informatica Cloud functionality. Can configure and test tasks, configure all related objects, and monitor jobs. Cannot perform administrative tasks for the organization.
### User Security Options

<table>
<thead>
<tr>
<th>User Security Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Note:</strong> Best practice is to assign one role to a user.</td>
<td></td>
</tr>
</tbody>
</table>
| User Groups | User group for the user. A user group can help define or limit the tasks that a user can perform. User groups are configured by the organization administrator. If no user groups are listed, no user groups are defined for the organization.  
**Note:** Best practice is to assign a maximum of one group to a user. Not all users need to be assigned to a user group. |
| Allow Data Preview | Allows the user to preview data when configuring a data loader or data synchronization task. By default, this option is enabled. For maximum data security, clear this option. |

5. Click **OK**.

**RELATED TOPICS:**
- “Organization Security Overview” on page 14

### Viewing User Details

You can view details about a user, such as the name, role, and contact information. When you view the details, you can click Edit to update user details.

- Click **Administration > Users**, and then click the user name.
  
  User details appear on the View User page.

### Deleting a User

Delete a user if the user no longer needs to access Informatica Cloud.

1. Click **Administration > Users**.
2. On the Users page, click the **Delete** icon next to the user name.

### Migrating Objects

You can migrate objects from one Informatica Cloud organization to another. To migrate objects, the target organization must have the Migration license.

You might need to migrate objects to another organization when you have multiple organizations. You can create multiple organizations to distinguish between environments.

For example, you can create separate organizations for development, test, and production environments. You create tasks in the development organization. After the tasks are developed, you can migrate the tasks to the test organization. After testing and validating all tasks in the test organization, you can migrate the tasks to the production organization.

After you migrate an object, the audit log displays a message indicating whether the migration was successful.

For information about obtaining the Migration license, contact Informatica Global Customer Support.
Migration Requirements

Before you migrate an object, ensure you meet the following requirements:

- You must have at least two organizations registered with Informatica Cloud.
- The target organization must have the Migration license.
- You have a user account for both organizations.
- Your user account in the target organization must have the Admin role.
- You created at least one Secure Agent in the organization where you want to migrate the object. If you do not create at least one Secure Agent, the following message appears when you try to migrate a task or connection:

  No Informatica Cloud Secure Agents available. You need to download and install a Secure Agent to migrate tasks.

Migrating an Object

You can migrate objects from a source organization to a target organization. A source organization is the organization that contains the object that you want to migrate. The target organization is the organization where you want to migrate the object.

You can migrate the following types of objects:

- Connections
- Data assessment tasks
- Data replication tasks
- Data synchronization tasks that do not include custom sources

When you select a task for migration, you must also select the corresponding source, target, and lookup connections if they do not exist in the target organization. To determine if a connection already exists, Informatica Cloud searches for a connection with the same name.

When you migrate a task, Informatica Cloud migrates the following components of the task:

- Connections
- Plug-ins
- Data filters
- Field expressions

When you migrate a task assigned to a schedule, Informatica Cloud migrates the task without the schedule. After you migrate the task, you can assign the task to a schedule defined in the target organization.

When you migrate a connection that is assigned to a Secure Agent in the source organization, Informatica Cloud fetches the list of Secure Agents from the target organization and assigns the migrated connection to the first Secure Agent in the list.

You cannot migrate a task if a task of the same name exists in the target organization. Rename the task if you want to migrate it. When you migrate a data replication task, Informatica Cloud does not change the target prefix.

To migrate an object:

1. Log in to the Informatica Cloud account of the target organization.
2. Click Administration > Migrate Tasks.
3. On the Migrate Tasks page, click Start Migration.
4. In the Login to Informatica Cloud Org area, click Authenticate.
5. In the Informatica Cloud Login dialog box, enter the user name and password of the Informatica Cloud source organization.
6. To migrate a task, in the Migrate Tasks area, click **Add**.
7. In the Add Tasks dialog box, select the type of task.
8. Add the tasks that you want to migrate to the Selected Tasks box, and click **OK**.
9. To add a connection, in the Migrate Connections area, click **Add**.
10. In the Add Connections dialog box, add the connections that you want to migrate to the Selected Connections box and click **OK**.
    The Existing Connections area shows connections already defined in the target organization.
11. To migrate the selected tasks and connections, click **OK**.

**Viewing the Audit Log**

The audit log displays Informatica Cloud events for each user. It displays user login, create, update, and delete operations.

By default, the audit log shows events for the past 12 months. To change the length of time that events appear in the audit log, contact Informatica Global Customer Support.

- Click **Administration > Audit Log**.
  The audit log appears on the Audit Log page.

**Billing Information**

When you subscribe to Informatica Cloud Express, you can view and edit billing information.

**Viewing Billing Information**

You can view billing information used to pay for Informatica Cloud Express on the Edit Billing page.

- Click **Administration > Billing**.
  The View Billing page displays the billing contact information and credit card details configured for the account.

**Editing Billing Information**

You can edit the billing name and address and the contact name and address associated with the account on the Edit Billing page. You can also view pricing information and frequently asked questions. To change the credit card type or number, contact Informatica Global Customer Support.

1. Click **Administration > Billing > Edit**.
   The Edit Billing page displays pricing information and frequently asked questions.
2. On the Edit Billing page, configure the following Billing Contact Detail fields:

<table>
<thead>
<tr>
<th>Billing Contact Details Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>First name.</td>
</tr>
<tr>
<td>Last Name</td>
<td>Last name.</td>
</tr>
<tr>
<td>Address</td>
<td>Full street address.</td>
</tr>
<tr>
<td>City</td>
<td>City.</td>
</tr>
<tr>
<td>State</td>
<td>State.</td>
</tr>
<tr>
<td>Postal Code</td>
<td>Postal or zip code.</td>
</tr>
<tr>
<td>Country</td>
<td>Country.</td>
</tr>
<tr>
<td>Phone</td>
<td>Phone number.</td>
</tr>
<tr>
<td>Email</td>
<td>Email address.</td>
</tr>
</tbody>
</table>

3. Click **OK**.

---

* Not configurable. To change the credit card type or number, contact Informatica Global Customer Support.
Organization Hierarchy

An organization hierarchy is a hierarchy of related organizations that includes a parent organization and one or more sub-organizations. To create an organization hierarchy, the parent organization must have the Org Hierarchy license.

Informatica Cloud provides the Informatica Cloud Administrator as a tool to create and manage organizations in the organization hierarchy. The organization administrator for a parent organization can use the Informatica Cloud Administrator.

For non-administrator users, an organization in a hierarchy functions like any stand-alone organization. To log in to an organization, all users require an Informatica Cloud account for the organization. A user in one organization in the hierarchy cannot log in to another organization in the hierarchy without a user account for that organization.

You can perform the following tasks with an organization hierarchy:

- Create a sub-organization
- Use the Informatica Cloud Administrator to manage the organization hierarchy

For information about obtaining the Org Hierarchy license, contact Informatica Global Customer Support.

Using the Informatica Cloud Administrator

The Informatica Cloud Administrator is a browser-based application that allows you to manage an Informatica Cloud organization hierarchy. Use the Informatica Cloud Administrator to create and manage sub-organizations, and to perform administrative tasks within the organizational hierarchy.

With the Informatica Cloud Administrator, you can perform administrative tasks such as editing organization properties, creating users, and viewing the Activity Log. If the organization has the Fine-Grained Security license, you can also configure user groups and object-level permissions.

You can use the Informatica Cloud Administrator if you are an administrator of an organization with the Org Hierarchy license.

You can perform the following tasks in the Informatica Cloud Administrator:

- **Activity log.** View the activity log and perform all related tasks, such as download log files.
- **Activity monitor.** View and refresh the activity monitor
- **Tasks.** View task details and configure object-level permissions.
- **Custom views.** Select and edit custom views.
- **Connections.** View connection details, edit and delete connections, and configure object-level permissions.
- **Schedules.** View schedule details. Create, edit, and delete schedules. Configure object-level permissions.
- **Task flows.** View task flow details, delete task flows, and configure object-level permissions.
- **Plug-ins.** View plug-in details, delete plug-ins, and configure object-level permissions.
- **Custom sources.** View custom source details, delete custom sources, and configure object level permissions.
- **My organization.** View and edit the parent organization properties and details.
- **Sub-organizations.** Create sub-organizations, view and edit sub-organization properties and details, and delete sub-organizations.
- **User groups.** View user group details. Create, edit, and delete user groups.
- **Users.** View user details. Create, edit, and delete users.
- **Audit log.** View the audit log.
Organization hierarchy. Change your view in the Informatica Cloud Administrator to other organizations in the hierarchy. Perform any of the listed tasks for sub-organizations.

The Informatica Cloud Administrator does not include all functionality offered in the Informatica Cloud application. For full Informatica Cloud functionality, use the Informatica Cloud application.

Logging in

An administrator of an organization with the Org Hierarchy license can log in to the Informatica Cloud Administrator. User groups and object-level permissions that restrict a user in the Informatica Cloud application also apply in the Informatica Cloud Administrator. Use your Informatica Cloud user account to log in to the Informatica Cloud Administrator.

Use the following URL to access the Informatica Cloud Administrator:

https://app.informaticaondemand.com/ac

Changing Between Organizations

You can change between organizations in the Informatica Cloud Administrator.

When you log in to the Informatica Cloud Administrator, the Organization field displays the name of the organization you are viewing.

To change to a different organization in the hierarchy, select the organization in the Organization list.

Creating a Sub-Organization

A sub-organization is an Informatica Cloud organization that is part of an organization hierarchy. You can create a sub-organization if you are the administrator of an organization with the Org Hierarchy license. Use the Informatica Cloud Administrator to create sub-organizations.

When you create a sub-organization, the sub-organization inherits all licenses and subscription options of the parent organization except for the Org Hierarchy license. A sub-organization cannot act as a parent organization for other organizations.

An organization hierarchy can include a limited number of sub-organizations. The Maximum Sub-Organizations organization property states the maximum number of sub-organizations that are allowed for your organization. To increase the number of sub-organizations allowed for your organization, contact Informatica Global Customer Support.

After you create a sub-organization, you can configure organization properties and details, as well as users and user groups. If you need to connect to the sub-organization through the Informatica Cloud application, create a new user account in the sub-organization.

To create a sub-organization:

1. Log in to the Informatica Cloud Administrator using your Informatica Cloud user account.

   Use the following URL to log in to the Informatica Cloud Administrator:

   https://app.informaticaondemand.com/ac

2. In the Informatica Cloud Administrator, click Administration > Sub-Organizations.

3. To create a new sub-organization, click New.

4. On the New Sub-Organization page, configure organization properties and details:
The following table describes organization properties:

<table>
<thead>
<tr>
<th>Organization Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization ID</td>
<td>Uniquely identifies the organization in the system. Informatica Cloud generates the organization ID.</td>
</tr>
<tr>
<td>Maximum Log Entries</td>
<td>Maximum number of entries that you want to list in the activity log. The maximum applies to all users in the organization that view the activity log. Enter an integer between 1 and 1000. Default is 100. You can clear the number of log entries on the View Organization page.</td>
</tr>
</tbody>
</table>

The following table describes organization details:

<table>
<thead>
<tr>
<th>Organization Detail</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization Name</td>
<td>Name of organization.</td>
</tr>
<tr>
<td>Address1</td>
<td>Address of organization.</td>
</tr>
<tr>
<td>Address2</td>
<td>Additional address information for the organization.</td>
</tr>
<tr>
<td>Address3</td>
<td>Additional address information for the organization.</td>
</tr>
<tr>
<td>City</td>
<td>City where the organization is based.</td>
</tr>
<tr>
<td>State</td>
<td>State where the organization is based.</td>
</tr>
<tr>
<td>Postal Code</td>
<td>Postal code of the area where the organization is based.</td>
</tr>
<tr>
<td>Country</td>
<td>Country where the organization is based.</td>
</tr>
<tr>
<td>Employees</td>
<td>Number of employees in the organization.</td>
</tr>
<tr>
<td>Do You Use Salesforce</td>
<td>Indicates whether your organization uses Salesforce.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the organization. Maximum length is 255 characters.</td>
</tr>
</tbody>
</table>

The following table describes default email notification options:

<table>
<thead>
<tr>
<th>Email Notification Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Email Notification</td>
<td>List of comma-separated email addresses that receive email notification when a task fails to complete. For example, <a href="mailto:admin@acme.com">admin@acme.com</a>, <a href="mailto:sa@acme.com">sa@acme.com</a>.</td>
</tr>
<tr>
<td>Warning Email Notification</td>
<td>List of comma-separated email addresses that receive email notification when a task completes with errors.</td>
</tr>
<tr>
<td>Success Email Notification</td>
<td>List of comma-separated email addresses that receive email notification when a task completes without errors.</td>
</tr>
</tbody>
</table>
The following table describes authentication options:

<table>
<thead>
<tr>
<th>Authentication Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Character Mix</td>
<td>Passwords can contain a mix of the following character sets: lowercase letters, capital letters, numbers, and special characters. Select the mix of characters that each password must contain:</td>
</tr>
<tr>
<td></td>
<td>- 1. Contains at least one of the character sets.</td>
</tr>
<tr>
<td></td>
<td>- 2. Contains at least two of the character sets.</td>
</tr>
<tr>
<td></td>
<td>- 3. Contains at least three of the character sets.</td>
</tr>
<tr>
<td></td>
<td>- 4. Contains all four character sets. Default is 1.</td>
</tr>
<tr>
<td>Password Reuse Allowed</td>
<td>Number of days until a previous password can be used again. Default is Always.</td>
</tr>
<tr>
<td>Password Expires</td>
<td>Number of days until the password expires. Default is Never.</td>
</tr>
<tr>
<td>Two-Factor Authentication</td>
<td>A security option that allows the configuration of trusted IP address ranges as an addition to account password authentication:</td>
</tr>
<tr>
<td></td>
<td>- Disabled. Informatica Cloud requires account passwords for access to the organization.</td>
</tr>
<tr>
<td></td>
<td>- Enabled. Informatica Cloud requires account passwords for access to the organization, and the user must login from within configured IP address ranges.</td>
</tr>
<tr>
<td>Trusted IP Ranges</td>
<td>Enabled when Two-Factor Authentication is selected. Enter one or more trusted IP address ranges. You can enter IP address ranges that use IP format version 4 (IPv4) or version 6 (IPv6). Use the Add icon to add another range. Use the Remove icon to remove a range.</td>
</tr>
</tbody>
</table>

5. Click **OK**.
CHAPTER 3

Configuring Informatica Cloud Components

This chapter includes the following topics:

- Informatica Cloud Secure Agent, 31
- Connections, 41
- Schedules, 54
- Task Flows, 58
- Plug-ins, 60
- Custom Sources, 63

Informatica Cloud Secure Agent

The Informatica Cloud Secure Agent is a lightweight program that runs all tasks and enables secure communication across the firewall between your organization and Informatica Cloud. When the Secure Agent runs a task, it connects to the Informatica Cloud hosting facility to access task information. Then the Secure Agent connects directly and securely to sources and targets, transfers data between sources and targets, and performs any additional task requirements.

When you install an Secure Agent, all Informatica Cloud users in the organization share the Secure Agent.

The Secure Agent must be running to run tasks. Run multiple Secure Agents to improve scalability or to create backup agents. To improve scalability, assign a different Secure Agent to each database or flat file target connection. You can run one Secure Agent on each machine.

Machine Requirements for the Secure Agent

The machine where you run the Secure Agent must meet the following requirements:

- Windows or Linux operating system (32-bit or 64-bit).
- 500 MB disk space.
- Internet access to the web server that runs the Informatica Cloud application.
- Network connectivity to source and target systems, if applicable.
Rules and Guidelines for the Secure Agent

Use the following rules and guidelines for the Secure Agent:

- The machine that runs the Secure Agent must meet the minimum requirements.
- You do not have to install the Secure Agent on the machine that you use to connect to Informatica Cloud. For example, you might use machine A to connect to Informatica Cloud and Machine B to run the Secure Agent.
- If you install the Secure Agent on Linux, do not include spaces in the directory path. If you include spaces, the Secure Agent installation may stop responding.
- If you install the Secure Agent on Windows, the Windows user account that installed the Secure Agent must have access to all remote directories that contain flat source or target files.
- You can run more than one Secure Agent, but run only one Secure Agent on each machine.
- The Secure Agent must be running to run tasks. If the Secure Agent loses connectivity to Informatica Cloud, it tries to reestablish connectivity to continue the task. If it cannot reestablish connectivity, the Secure Agent becomes disabled and the task fails.
- A Secure Agent that runs on Linux does not support all combinations of sources and targets. A Secure Agent that runs on Linux supports the following combinations of sources and targets in a task:
  - Salesforce source with a flat file, MySQL, Oracle, or Salesforce target.
  - Flat file, MySQL, or Oracle source with a Salesforce target.
- A Secure Agent that runs on Linux does not support tasks that contain a Microsoft SQL Server source or target.
- A Secure Agent that runs on Windows supports any combination of sources and targets that the task supports.
- If an Informatica Cloud service shuts down for more than 12 hours, the Secure Agent connection to the Informatica Cloud service times out. Manually restart the Secure Agent on each client machine where it is installed.
- You must uninstall the Secure Agent before you install another Secure Agent on the same machine.

Working with the Secure Agent and Secure Agent Manager on Windows

On Windows, the Secure Agent runs as a Windows service. When you install the Secure Agent, you also install the Informatica Cloud Secure Agent Manager.

By default, the Secure Agent starts when you start Windows. Use the Secure Agent Manager to stop and restart the Secure Agent. You can also use the Secure Agent Manager to check the Secure Agent status, and configure proxy information.

You can launch the Secure Agent Manager from the Start Menu or desktop icon.

When you close the Secure Agent Manager, it minimizes to the Windows taskbar notification area for quick access. Closing the Secure Agent Manager does not affect the Secure Agent status.

To configure a Secure Agent:

1. Ensure the machine on which you run the Secure Agent meets the minimum requirements.
2. Download the Secure Agent installation wizard.
3. Install and register the Secure Agent.

You can perform the following tasks for a Secure Agent:

- View the Secure Agent status.
- View the Secure Agent details.
- View the Secure Agent history.
Stop the Secure Agent.
- Restart the Secure Agent.
- Configure proxy settings.
- Configure a Windows network login.
- Delete a Secure Agent.
- Uninstall a Secure Agent.

**Downloading the Secure Agent Installation Wizard**

Download the Secure Agent installation wizard from Informatica Cloud. You can download and run the Secure Agent on any machine that meets the minimum requirements.

1. Click **Configuration > Secure Agents**, and then click **Download Secure Agent**.
2. Select the operating system on which you will run the Secure Agent, and then click **Download**.
3. Save the installation wizard to the machine where you want to run the Secure Agent.

   The name of the Windows Secure Agent installation file is `agent_install.exe`. The name of the Linux Secure Agent installation file is `agent_install.bin`.

**Installing the Secure Agent on Windows**

You can install one Secure Agent on each machine. You must uninstall the existing Secure Agent from the machine before you install a new Secure Agent. After the installation completes, a registration page appears. Use your Informatica Cloud username and password to register the Secure Agent. After you complete the registration, the Secure Agent Manager displays the Secure Agent status.

If the organization requires a proxy server to connect to the internet, the installation wizard configures the proxy settings for the Secure Agent based on the proxy settings in the browser. You must configure the proxy settings if they change or are incorrectly configured by the installation program.

Configure a login to allow the Secure Agent to access directories to configure and run tasks. When you configure or run tasks that use Flat File or FTP/SFTP connection types, the Secure Agent might require permissions to read or write information to related directories.

To install the Secure Agent on Windows:

1. Open the directory where you downloaded `agent_install.exe`, and double-click the file.
2. Choose the installation folder and click **Next**.
3. Review the pre-installation summary and click **Install**.
4. After the installer completes, click **Done**.

   A registration page appears.

5. Enter your Informatica Cloud user name and password and click **Register**.

   The Secure Agent starts.

   The Informatica Cloud Secure Agent window displays the status of the Secure Agent. You can restart, stop, and configure the Secure Agent proxy in this window. You can close the window at any time. The Secure Agent continues to run as a service until stopped.

6. Close the Secure Agent Manager.

   The Secure Agent Manager minimizes to the Windows taskbar notification area. Closing the Secure Agent Manager does not affect the Secure Agent status.
**View the Secure Agent Status from the Secure Agent Manager**

The Secure Agent Manager displays the Secure Agent status.

To view the Secure Agent status:

- To launch the Secure Agent Manager, click **Start > All Programs > Informatica Cloud Secure Agent > Informatica Cloud Secure Agent**.
  
  If the Secure Agent Manager is active, you can hover over the Informatica Cloud icon to view the Secure Agent status. Or, you can click the Informatica Cloud icon in the Windows taskbar notification area to open the Secure Agent Manager.
  
  The Secure Agent Manager displays the Secure Agent status.

**View the Secure Agent Status from Informatica Cloud**

You can view the status of a Secure Agent on the Configuration tab of Informatica Cloud. If the status is inactive, the Secure Agent is not running. If the status is active, it is running and you can use it to run tasks.

**Note:** After the Secure Agent starts, you might need to wait for the Secure Agents page to update the status. The response time depends on network performance. The Secure Agents page refreshes every five seconds to display the current status.

To view the status of a Secure Agent:

- Click **Configuration > Secure Agents**.
  
  The status appears on the Secure Agents page.

**View Secure Agent Details**

You can view detailed information about the Secure Agent on the View Secure Agents page. On the View Secure Agents page, you can also refresh the status of the Secure Agent.

To view Secure Agent details:

1. Click **Configuration > Secure Agents**.
   
   The Secure Agents page lists all Secure Agents for your organization.

2. Click the name of the Secure Agent that you want to view.
   
   The View Secure Agent page displays Secure Agent details and version.

3. To refresh the status of the Secure Agent, click **Refresh Status**.

**View the Secure Agent History**

You can view the Secure Agent history. The Secure Agent history includes start and stop times, server connection and upgrade information.

You can configure the number of rows displayed on the page.

To view the Secure Agent history:

1. Click **Configuration > Secure Agents**.

2. On the Secure Agents page, click the name of the Secure Agent.

3. On the View Secure Agent page, click **View Secure Agent History**.
   
   The Secure Agent History page appears.

4. Use the **Rows per Page** option to change the number of rows that display on each page.
Stopping and Restarting the Secure Agent on Windows

On Windows, the Secure Agent installation wizard starts the Secure Agent service. The installation wizard also configures the Secure Agent service to run upon Windows startup.

You can use the Secure Agent Manager to stop or restart the Secure Agent.

1. To launch the Secure Agent Manager, click Start > All Programs > Informatica Cloud Secure Agent > Informatica Cloud Secure Agent.
   If the Secure Agent Manager is active, you can click the Informatica Cloud icon in the Windows taskbar notification area to open the Secure Agent Manager.
   The Secure Agent Manager displays the Secure Agent status.

2. To stop the Secure Agent, click Stop. To restart the Secure Agent, click Restart.
   The Secure Agent Manager displays a message when the action is complete.

3. Close the Secure Agent Manager.
   The Secure Agent Manager minimizes to the Windows taskbar notification tray. Closing the Secure Agent Manager does not affect the Secure Agent status.

Configuring the Proxy Settings on Windows

If your organization uses an outgoing proxy server to connect to the internet, the Secure Agent connects to the Informatica Cloud application through the proxy server. The Secure Agent installation wizard configures the proxy server settings for the Secure Agent based on settings configured in the browser. Configure proxy server settings if they change or if they are incorrectly configured by the installation program.

Contact your network administrator for the correct proxy settings.

To configure the proxy settings for the Secure Agent on a Windows machine:

1. To launch the Secure Agent Manager, click Start > All Programs > Informatica Cloud Secure Agent > Informatica Cloud Secure Agent.
   If the Secure Agent Manager is active, you can click the Informatica Cloud icon in the Windows taskbar notification area to open the Secure Agent Manager.
   The Secure Agent Manager displays the Secure Agent status.

2. In the Secure Agent Manager, click Proxy.

3. To enter proxy server settings, click Use a Proxy Server.

4. Enter the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy Host</td>
<td>Required</td>
<td>Host name of the outgoing proxy server that the Secure Agent uses.</td>
</tr>
<tr>
<td>Proxy Port</td>
<td>Required</td>
<td>Port number of the outgoing proxy server.</td>
</tr>
<tr>
<td>User Name</td>
<td>Optional</td>
<td>User name to connect to the outgoing proxy server.</td>
</tr>
<tr>
<td>Password</td>
<td>Optional</td>
<td>Password to connect to the outgoing proxy server.</td>
</tr>
</tbody>
</table>

5. Click OK.
   The Secure Agent Manager restarts the Secure Agent to apply the new settings.
Configuring a Login for a Windows Secure Agent Service

On Windows, you can configure a network login for the Secure Agent service. A login allows the Secure Agent to access the network with the privileges and permissions associated with the login.

Configure a login to allow the Secure Agent to access directories to configure and run tasks. When you configure or run tasks that use Flat File or FTP/SFTP connection types, the Secure Agent might require permissions to read or write information to related directories.

For example, to browse to a directory to configure a Flat File or FTP/SFTP connection, the Secure Agent login might require permission to access the directory. Without a Secure Agent login with the appropriate permissions, Informatica Cloud cannot display the directory in the Browse for Directory dialog box.

To configure a login for a Secure Agent service:

1. From the Windows Control Panel, click Administrative Tools > Services.
2. In the Services window, right-click the Informatica Cloud Secure Agent service and click Properties.
3. In the properties dialog box, click the Log On tab.
4. To configure a login, select This Account.
5. Enter an account and associated password.
   Use an account with the required privileges and permissions for the network security defined for the domain. By default, the account format is <domain_name>\<user_name>.
6. Click OK.
7. In the Services window, restart the Secure Agent service for the changes to take effect.

Deleting a Secure Agent

Delete a Secure Agent if you no longer need it to run tasks.

You cannot delete a Secure Agent if it is associated with a connection. Before you delete a Secure Agent, update associated connections to use another Secure Agent.

1. Click Configuration > Secure Agents.
2. On the Secure Agents page, click the Delete icon next to the Secure Agent.
   If the Secure Agent is running, a warning message appears. Stopping an active Secure Agent prevents scheduled tasks associated with the Secure Agent from running. Ignore the warning if you do not need the Secure Agent.

If you no longer need the Secure Agent, uninstall the Secure Agent.

Uninstalling the Secure Agent on Windows

You can uninstall the Secure Agent. You might uninstall the Secure Agent if you no longer want to run the Secure Agent on the machine or if you want to reinstall the Secure Agent.

Before you uninstall the Secure Agent, make sure no connections are configured to use it.

To uninstall the Secure Agent:

1. Click Start > All Programs > Informatica Cloud Secure Agent > Uninstall Informatica Cloud Secure Agent.
   The Secure Agent uninstaller launches.
2. Click Uninstall.
3. When the uninstall completes, click Done.
4. Delete any remaining files in the installation directory.
    After you uninstall the Secure Agent, log files may still exist in the directory where you installed the Secure Agent.

**Working with the Secure Agent on Linux**

On Linux, the Secure Agent runs as a process. You can use a shell command line to install, register, start, stop, and uninstall the Secure Agent. You can also use the shell command line to check the Secure Agent status. Once started, the Secure Agent continues to run as a process until stopped.

Note the following limitations when using the Secure Agent on Linux:

- A Secure Agent that runs on Linux does not support all combinations of sources and targets. A Secure Agent that runs on Linux supports the following combinations of sources and targets in a task:
  - Salesforce source with a flat file, MySQL, Oracle, or Salesforce target.
  - Flat file, MySQL, or Oracle source with a Salesforce target.
- A Secure Agent that runs on Linux does not support tasks that contain a Microsoft SQL Server source or target.

To configure a Secure Agent:

1. Ensure the machine on which you run the Secure Agent meets the minimum requirements.
2. Download the Secure Agent installation wizard.
3. Install, start, and register the Secure Agent.

You can perform the following tasks for a Secure Agent:

- View the Secure Agent status.
- View the Secure Agent details.
- View the Secure Agent history.
- Start and stop the Secure Agent.
- Configure proxy settings.
- Delete a Secure Agent.
- Uninstall the Secure Agent.

**Downloading the Secure Agent Installation Wizard**

Download the Secure Agent installation wizard from Informatica Cloud. You can download and run the Secure Agent on any machine that meets the minimum requirements.

1. Click **Configuration > Secure Agents**, and then click **Download Secure Agent**.
2. Select the operating system on which you will run the Secure Agent, and then click **Download**.
3. Save the installation wizard to the machine where you want to run the Secure Agent.
   
   The name of the Linux Secure Agent installation file is `agent_install.bin`.

**Installing and Registering the Secure Agent on Linux**

You can install one Secure Agent on each machine. You must uninstall the existing Secure Agent from the machine before you install a new Secure Agent. After the installation completes, start the Secure Agent, then use your Informatica Cloud username and password to register the Secure Agent.
If the organization requires a proxy server to connect to the internet, the installation wizard configures the proxy settings for the Secure Agent based on the proxy settings in the browser. You must configure the proxy settings if they change or are incorrectly configured by the installation program.

To install the Secure Agent on Linux:

1. In a shell command line, navigate to the directory where you downloaded the file and enter the following command:
   
   `agent_install.bin -i console`

   The installer completes.

2. To start the Secure Agent, navigate to the directory where you installed the Secure Agent and enter the following command:
   
   `infaagent startup`

   After the Secure Agent starts, register the Secure Agent.

3. To register the Secure Agent, navigate to the following directory:
   
   `<SecureAgent_InstallDir>/main/agentcore`

4. Enter the following command using your Informatica Cloud username and password:
   
   `consoleAgentManager.sh configure <username> <password>`

   You can check the registration status of a Secure Agent using the following command in the same directory:

   `consoleAgentManager.sh isConfigured`

**View the Secure Agent Status on Linux**

To view the Secure Agent status on Linux:

1. In a shell command line, navigate to the following directory:
   
   `<SecureAgent_InstallDir>/main/agentcore`

2. Run the following command:
   
   `consoleAgentManager.sh getstatus`

   The command returns the Secure Agent status.

3. To refresh the status of the Secure Agent, you can repeat Step 2, or you can run the following command:
   
   `consoleAgentManager.sh updatestatus`

   You might need to use the updatestatus command if you have intermittent connectivity with the Secure Agent.

**View the Secure Agent Status from Informatica Cloud**

You can view the status of a Secure Agent on the Configuration tab of Informatica Cloud. If the status is inactive, the Secure Agent is not running. If the status is active, it is running and you can use it to run tasks.

**Note**: After the Secure Agent starts, you might need to wait for the Secure Agents page to update the status. The response time depends on network performance. The Secure Agents page refreshes every five seconds to display the current status.

To view the status of a Secure Agent:

- Click **Configuration > Secure Agents**.

  The status appears on the Secure Agents page.
View Secure Agent Details

You can view detailed information about the Secure Agent on the View Secure Agents page. On the View Secure Agents page, you can also refresh the status of the Secure Agent.

To view Secure Agent details:

1. Click **Configuration > Secure Agents**.
   - The Secure Agents page lists all Secure Agents for your organization.
2. Click the name of the Secure Agent that you want to view.
   - The View Secure Agent page displays Secure Agent details and version.
3. To refresh the status of the Secure Agent, click **Refresh Status**.

View the Secure Agent History

You can view the Secure Agent history. The Secure Agent history includes start and stop times, server connection and upgrade information.

You can configure the number of rows displayed on the page.

To view the Secure Agent history:

1. Click **Configuration > Secure Agents**.
2. On the Secure Agents page, click the name of the Secure Agent.
3. On the View Secure Agent page, click **View Secure Agent History**.
   - The Secure Agent History page appears.
4. Use the **Rows per Page** option to change the number of rows that display on each page.

Edit the Secure Agent

To edit the Secure Agent:

1. Click **Configuration > Secure Agents**.
2. On the Secure Agents page, click the name of the Secure Agent.
3. On the View Secure Agents page, click **Edit**.
4. For **Secure Agent Name**, enter the name that you want to use, and click **OK**.

Starting and Stopping the Secure Agent on Linux

After you download the Secure Agent program files on a Linux machine, you can run the Secure Agent as a Linux process. Manually start the Secure Agent process on Linux.

1. From the command line, navigate to the directory where you installed the Secure Agent.
2. To start the Secure Agent, enter the following command:
   ```
   infaagent startup
   ```
3. To stop the Secure Agent, enter the following command:
   ```
   infaagent shutdown
   ```

You can view the Secure Agent status from Informatica Cloud or from a Linux command line.
Configuring the Proxy Settings on Linux

If your organization uses an outgoing proxy server to connect to the internet, the Secure Agent connects to the Informatica Cloud application through the proxy server. The Secure Agent installation wizard configures the proxy server settings for the Secure Agent based on settings configured in the browser. If the proxy server settings are not set correctly or change, update the proxy server settings defined for the Secure Agent.

Contact the network administrator to determine the proxy settings.

To configure the proxy server settings for the Secure Agent on a Linux machine:

1. Open the following directory:
   `<SecureAgent_InstallDir>\main`

2. Edit the following properties in the `proxy.ini` file:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InfaAgent.ProxyHost</td>
<td>Host name of the outgoing proxy server that the Secure Agent uses.</td>
</tr>
<tr>
<td>InfaAgent.ProxyPort</td>
<td>Port number of the outgoing proxy server.</td>
</tr>
<tr>
<td>InfaAgent.ProxyUser</td>
<td>User name to connect to the outgoing proxy server.</td>
</tr>
<tr>
<td>InfaAgent.ProxyPassword</td>
<td>Password to connect to the outgoing proxy server.</td>
</tr>
</tbody>
</table>

3. Save and close the file.
4. Restart the Secure Agent.

Deleting a Secure Agent

Delete a Secure Agent if you no longer need it to run tasks.

You cannot delete a Secure Agent if it is associated with a connection. Before you delete a Secure Agent, update associated connections to use another Secure Agent.

1. Click Configuration > Secure Agents.
2. On the Secure Agents page, click the Delete icon next to the Secure Agent.
   
   If the Secure Agent is running, a warning message appears. Stopping an active Secure Agent prevents scheduled tasks associated with the Secure Agent from running. Ignore the warning if you do not need the Secure Agent.

If you no longer need the Secure Agent, uninstall the Secure Agent.

Uninstalling the Secure Agent on Linux

You can uninstall the Secure Agent. You might uninstall the Secure Agent if you no longer want to run the Secure Agent on the machine or if you want to reinstall the Secure Agent.

Before you uninstall the Secure Agent, make sure no connections are configured to use it.

To uninstall the Secure Agent on Linux:

1. Open the command line.
2. Navigate to the directory where you installed the Secure Agent.
3. Enter the following command in the command line to stop the Secure Agent Linux process:
   
   `infaagent shutdown`
4. To uninstall the Secure Agent, run `rm -rf` on the directory where you installed the Secure Agent to remove Secure Agent files.

## Connections

A connection is an Informatica Cloud object that you configure to connect to Salesforce, SAP, web services, databases, and flat files. You use connections to specify the sources and targets for tasks.

The following table lists all possible connection types for Informatica Cloud Services:

### Table 1. Connection Types for Data Services

<table>
<thead>
<tr>
<th>Connection Type</th>
<th>Applicable Informatica Cloud Services</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat file</td>
<td>Data Loader service</td>
<td>Stores information required to create, access, and store flat files. It includes the directory and data format for flat files.</td>
</tr>
<tr>
<td></td>
<td>Data Replication service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Synchronization service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PowerCenter service</td>
<td></td>
</tr>
<tr>
<td>FTP/SFTP</td>
<td>Data Synchronization service</td>
<td>Stores information required to create, access, and store flat files using FTP or SFTP. It includes the remote directory and a local directory for staged local files.</td>
</tr>
<tr>
<td></td>
<td>PowerCenter service</td>
<td></td>
</tr>
<tr>
<td>SQL Server</td>
<td>Data Loader service</td>
<td>Stores information required to connect to Microsoft SQL Server databases, version 2000, 2005, and 2008. It includes the database name, login, and compatible code page.</td>
</tr>
<tr>
<td></td>
<td>Data Replication service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Synchronization service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PowerCenter service</td>
<td></td>
</tr>
<tr>
<td>MS Access</td>
<td>Data Synchronization service</td>
<td>Stores information required to connect to a Microsoft Access database. It includes the system DSN and compatible code page.</td>
</tr>
<tr>
<td>MySQL</td>
<td>Data Loader service</td>
<td>Stores information required to connect to a MySQL database, version 5.0.x. It includes the login, database information, and compatible code page.</td>
</tr>
<tr>
<td></td>
<td>Data Replication service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Synchronization service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PowerCenter service</td>
<td></td>
</tr>
<tr>
<td>ODBC</td>
<td>Data Synchronization service</td>
<td>Stores information required to connect to a database using ODBC. It includes a system DSN, login, and compatible code page. Use an ODBC connection when Informatica Cloud does not provide a connection for the database or version you want to use.</td>
</tr>
<tr>
<td>Oracle</td>
<td>Data Loader service</td>
<td>Stores information required to connect to an Oracle database, version 10g or 11. It includes the login, service name, and compatible code page.</td>
</tr>
<tr>
<td></td>
<td>Data Replication service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Synchronization service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PowerCenter service</td>
<td></td>
</tr>
<tr>
<td>Salesforce</td>
<td>Data Assessment service</td>
<td>Stores information required to connect to a Salesforce account, API version 18 or 19. Default is 19. It includes the Salesforce login and service URL. You can create a connection to any type of Salesforce account. You can create connections to the following Salesforce editions: Professional Edition, Enterprise Edition, and Unlimited Edition.</td>
</tr>
<tr>
<td></td>
<td>Data Loader service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Replication service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Data Synchronization service</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PowerCenter service</td>
<td></td>
</tr>
<tr>
<td>Connection Type</td>
<td>Applicable Informatica Cloud Services</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SAP</td>
<td>PowerCenter Service</td>
<td>Stores information required to use an SAP IDoc Reader or Writer to access SAP data. An SAP IDoc Reader connection includes the destination entry and compatible code page. An SAP IDoc Writer connection includes the connection string, language and client codes, and compatible code page.</td>
</tr>
<tr>
<td>Web Service</td>
<td>PowerCenter Service</td>
<td>Stores information required to connect to web service data. It includes the endpoint URL, and optional trusted certificates and private key file information.</td>
</tr>
</tbody>
</table>

You can perform the following tasks for connections:

- Create a connection.
- Edit a connection.
- View connection details.
- Copy a connection.
- Test a connection.
- Delete a connection.

**Understanding FTP/SFTP Connections**

File Transfer Protocol (FTP) connections allow you to use FTP to access source and target files. Secure File Transfer Protocol (SFTP) connections use secure protocols, such as SSH, to access source and target files.

When you configure an FTP/SFTP connection, you define a remote directory and a local directory. The remote directory is the location of the files you want to use as sources or targets. The local directory is a directory local to the Secure Agent that contains a copy of the source or target files.

When you work with FTP/SFTP connections, make sure the local directory contains valid copies of all source and target files. When you configure a task with an FTP/SFTP connection, Informatica Cloud uses the file structure of the local file to define the source or target for the task. The file structure of the local file must match the source or target file in the remote directory.

Informatica Cloud also uses the local file to generate data preview. If the data in the local file does not match the data in the source or target file in the remote directory, data preview can display inaccurate results.

When Informatica Cloud runs a task with a FTP/SFTP target connection, it creates a target file based on the target defined in the task. As it completes the task, Informatica Cloud writes the target file to the remote directory, overwriting the existing file.

**FTP/SFTP Connections for PowerCenter Tasks**

If you create a PowerCenter task with an FTP/SFTP target connection and the IS_STAGED option is enabled for the underlying PowerCenter session, Informatica Cloud writes the flat file to the remote machine and the following local directory:

```
<SecureAgent_InstallDirectory>\main\rdtmDir
```

For PowerCenter tasks, Informatica Cloud ignores the Local Directory property specified in the FTP/SFTP connection. Instead, it uses properties specified in the PowerCenter session. To change the local directory or default local filename, change the Output File Directory and Output Filename session properties in PowerCenter, export the workflow from PowerCenter to an XML file, and re-import the XML file into Informatica Cloud.
Understanding ODBC Connections

You can create an ODBC connection to connect to a database. Before you configure the ODBC connection in Informatica Cloud, you must create a system Data Source Name (DSN). When you create the system DSN, you must specify the data source name, connect string, and choose a database driver that is compatible with the database to which you want to connect.

An ODBC connection supports only the functionality available through the ODBC driver for the source or target you want to use. For example, ODBC drivers do not allow updates or deletes of Excel data, so you cannot use Informatica Cloud to update, upsert, or delete data in Excel targets.

If you use the MySQL ODBC 3.51 driver for an ODBC connection and you select a MySQL target that has a column with a Double datatype, an error similar to the following error appears:

```
Column [A1_NUMBER_18_0] has scale greater than precision.
```

To resolve the error, upgrade to the MySQL ODBC 5.1 driver.

Understanding Salesforce Connections

Salesforce connections allow you to securely read data from or write data to Salesforce sources or targets. You can use Salesforce connections in any Informatica Cloud task. Data Loader tasks must include at least one Salesforce connection.

Security Tokens and Trusted IP Ranges

Most Salesforce accounts require a security token to access the account. When you create a Salesforce connection, you can enter the security token. If your account requires a security token and you do not have one, you can generate a security token. After you log in at the Salesforce web site, click Setup > My Personal Information > Reset My Security Token.

To avoid adding the security token to a Salesforce connection, you can add Informatica Cloud IP addresses to the Trusted IP Ranges in your Salesforce account. At the Salesforce web site, click Setup > Security Controls > Network Access, then add the following IP addresses:

- Informatica Cloud. Add range 209.135.42.0 - 209.135.42.255.
- Secure Agent machines. Add individual IP addresses or range of all machines that run a Secure Agent.

For more information, see the Salesforce documentation.

Understanding SAP Connections

SAP connections allow you to securely read and write SAP IDoc data. Use SAP connections in PowerCenter tasks. Use SAP connections for PowerCenter tasks that include Informatica PowerCenter SAPALEIDoc sources or targets.

Informatica Cloud provides two types of SAP connections:

- iDoc Reader. Use to read from SAP IDoc sources.
- iDoc Writer. Use to write to SAP IDoc targets.

Understanding Web Service Connections

Web Services connections allow you to securely read data from or write data to web service sources or targets. Use a Web Service connection for plug-ins or PowerCenter tasks.
To connect to a web service, Informatica Cloud requires an endpoint URL. If you configure a Web Service connection without providing an endpoint URL, Informatica Cloud uses the endpoint URL contained in the WSDL file on which the source or target is based.

**Rules and Guidelines for Connections**

Use the following rules and guidelines all types of connections:

- If the source data is of a different code page than the one specified for the target, Informatica Cloud may load unexpected data into the target.
- If you select a Secure Agent that runs on Linux, you cannot specify a Windows directory for a flat file target.
- When you create a source connection to a database, you can create the connection to a database table, alias, or view. You can also create a source connection to an Oracle synonym. When you create a target connection to a database, you can create a connection to a database table.

**Rules and Guidelines for Flat File Connections**

Use the following rules and guidelines when you create a flat file connection:

- You cannot use a flat file in a data synchronization task if the flat file name contains one of the following characters:
  - `&`
  - `\`
- On Windows, you cannot select flat file directory on a mapped drive through the Browse for Directory dialog box. You can access a network directory by browsing My Network Places. You can also enter the directory with the following format:
  \<server_name>\<directory_path>
  
  If the Browse for Directory dialog box does not display My Network Places, you might need to configure a network login for the Secure Agent service.

**Rules and Guidelines for FTP/SFTP Connections**

Use the following rules and guidelines for FTP/SFTP connections:

- Informatica Cloud does not lock the target file while writing to the file. To prevent data corruption, verify that only one task writes to a target file at any given time.
- Informatica Cloud validates the flat file in the local directory, not the remote directory.
- If metadata in the local target file and remote target file are different, Informatica Cloud overwrites the metadata of the remote target file with the local target file at runtime.
- The Activity Log shows the row count of rows loaded into the local target file.
- On Windows, you cannot select FTP/SFTP directory on a mapped drive through the Browse for Directory dialog box. You can access a network directory by browsing My Network Places. You can also enter the directory with the following format:
  \<server_name>\<directory_path>
  
  If the Browse for Directory dialog box does not display My Network Places, you might need to configure a network login for the Secure Agent service.
- Error messages for FTP/SFTP connections might only reference FTP or SFTP. Read any error message that references FTP or SFTP as an error message for an FTP/SFTP connection.
Rules and Guidelines for MS Access Connections

Use the following rules and guidelines when you create a MS Access connection:

- You can only use the insert task operation for tasks that contain an MS Access connection and a lookup defined in a field mapping. The task will fail for all other task operations.
- You cannot use multiple source objects in a data synchronization task when the task contains an MS Access source or target that uses primary and foreign keys.
- The Microsoft Access connection does not support Unicode (UTF-8) data. No tasks can read or write Unicode data from a Microsoft Access table.

Rules and Guidelines for ODBC Connections

Use the following rules and guidelines when you create an ODBC connection:

- Use uppercase letters when you specify the schema name for an Oracle database that uses an ODBC connection.
- ODBC connections support System DSNs, not User DSNs.
- Do not use an ODBC connection for databases that have predefined connection types. For example, use the Oracle connection type to connect to an Oracle database.
- You cannot use a Secure Agent that runs on Linux for an ODBC connection. Use a Secure Agent that runs on Windows.
- You can use ODBC connections in data loader and data synchronization tasks. You cannot create or edit ODBC connections in the Data Loader Task Wizard.
- When you create or edit a task with an ODBC connection, database tables from other schema in the database may appear in the wizard. The wizard does not filter tables based on the schema specified for the ODBC connection.
- ODBC connections do not support Unicode (UTF-8) data. No tasks can read or write Unicode data using an ODBC connection.
- The data preview area may not display Unicode characters of a source or target that uses an ODBC connection.
- The data preview area may not display data from an ODBC connection if the database table or column name is also a database key word.
- If you use an ODBC connection for an Oracle database target, ensure Oracle table columns with the following datatypes do not exceed the specified maximum precision: char(1999), varchar(3999), nvarchar(3998), and nchar(3998).
- If you use an ODBC connection for an Excel target, data is inserted into the target. The ODBC driver does not allow you to update or delete existing Excel data.
- If you use an ODBC connection for an Excel source or target file, ensure named ranges are defined in the Excel file.

Rules and Guidelines for Oracle Connections

Use the following rules and guidelines when you create an Oracle connection:

- When you run a task that contains an Oracle connection and the Oracle database is in the Public schema, the task may time out if the schema contains too many objects. To resolve the error, remove some objects from the Oracle database or move the objects into another database schema.
- When you run a task that contains an Oracle connection for an Oracle database target, Informatica Cloud may truncate UTF-8 characters if they exceed the maximum length of the varchar or char fields. The ODBC driver allocates 6 bytes per character.
Rules and Guidelines for Salesforce Connections
Use the following rules and guidelines when you create a Salesforce connection:

- When you create a Salesforce connection, you specify the user name and password of the Salesforce account. You may have to append the Salesforce security token to the password to connect to the Salesforce account.

Understanding the Service URL
The service URL is the uniform resource locator for a source or target connection. It appears in the list of connections on the Connections page.

The service URL differs based on the following types of connections:

- Flat file connection.
  The service URL is the directory that stores the flat files.

- Salesforce connection.
  The service URL is the Salesforce service URL that you access.

- Database connection.
  The service URL consists of the host, port, and service name that you configure in the connection. It uses the following format:
    `<host>:<port>/<service_name>`

Task Operations
When you configure the data loader or data synchronization task, you specify the type of target and task operation. The available task operations depend on the target type that you select.

The following table lists the task operations available for each target type:

<table>
<thead>
<tr>
<th>Target Type</th>
<th>Available Task Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat file</td>
<td>Insert</td>
</tr>
<tr>
<td>Database</td>
<td>Insert, Delete, Update, or Upsert</td>
</tr>
<tr>
<td>Salesforce</td>
<td>Insert, Delete, Update, or Upsert</td>
</tr>
</tbody>
</table>

Insert Task Operation
When you run a task with the Insert task operation, Informatica Cloud inserts all source rows into the target. If Informatica Cloud finds a source row that exists in the target, the row fails. The activity log and activity monitor display the number of failed rows.

If you write data to a flat file target, Informatica Cloud truncates the flat file before it inserts the source rows into the file.

Update Task Operation
When you run a task with the Update task operation, Informatica Cloud updates rows in the target that exist in the source. If Informatica Cloud finds a row in the source that does not exist in the target, the row fails. The activity log and activity monitor display the number of failed rows.
Upsert Task Operation

When you run a task with the Upsert task operation, Informatica Cloud updates all rows in the target that also exist in the source and inserts all new source rows into the target.

If a source field contains a NULL value and the corresponding target field contains a value, Informatica Cloud retains the existing value in the target field.

If the task uses the Upsert task operation on a Salesforce target, you must either map the Salesforce ID or the external ID to identify existing records in the target.

**Note:** The task fails if the Salesforce user account defined for the task does not have creatable or updatable permissions on the external ID field. When using the upsert task operation with an External Id field that is write-protected, ensure that the external IDs exist.

Delete Task Operation

When you run a data loader task with the Delete task operation, the Data Loader service deletes all rows from the target that exist in the source.

Configuring a Connection

You can configure a connection on the Configuration tab or in a wizard as you configure a task. In both cases, the connection becomes available to the entire organization to use. When you configure a connection in a task, the connection dialog box displays only valid connection types.

For all connection types except Salesforce, install at least one Secure Agent before you configure the connection. When you create one of these connections, you specify the Secure Agent. The Secure Agent must be installed and running when you configure the connection.

Edit a connection to update the connection properties. You might edit the connection properties to change the source or target object that the connection includes or to change the Secure Agent associated with the connection.

To configure a connection:

1. To create a connection, click **Configuration > Connections**, and then click **New**.
   - To edit an existing connection, click **Configuration > Connections**, click the connection name, and then click **Edit**.

2. On the New Connection or Edit Connection page, configure the following connection details:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
   | Connection Name | Name of the connection. Each connection name must be unique within the organization. Connection names can contain alphanumeric characters, spaces, and the following special characters: _ - + .  
   | Description   | Connection names are not case sensitive.                                    |
   | Type          | Description of the connection. Maximum length is 255 characters.            |
   | Type          | Type of connection.                                                         |

3. Configure additional connection details for the source or target connection you want to create:
For Flat File connections:

<table>
<thead>
<tr>
<th>Flat File Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Secure Agent that Informatica Cloud uses to access the flat file in the local area network.</td>
</tr>
</tbody>
</table>
| Directory       | Directory where the flat file is stored. Enter the full directory or click Browse to locate and select the directory. Maximum length is 100 characters. Directory names can contain alphanumeric characters, spaces, and the following special characters: / \ : _ ~. On Windows, the Browse for Directory dialog box does not display mapped drives. You can browse My Network Places to locate the directory or enter the directory with the following format: \\
<server_name> \\
<directory_path>. If network directories do not display, you can configure a login for the Secure Agent service to resolve the issue. Do not include the name of the flat file. You specify the file name when you create the task. |
| Browse button   | Use to locate and select the directory where flat files are stored. |
| Date Format     | Date format for date fields in the flat file. Default date format is: MM/dd/yyyy HH:mm:ss |
| Code Page       | The code page of the system that hosts the flat file. Select one of the following code pages: - MS Windows Latin 1. Select for ISO 8859-1 Western European data. - UTF-8. Select for Unicode data. |

For FTP/SFTP connections:

<table>
<thead>
<tr>
<th>FTP/SFTP Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Secure Agent that Informatica Cloud uses to access the flat file in the local area network.</td>
</tr>
<tr>
<td>User Name</td>
<td>User name for the database login.</td>
</tr>
<tr>
<td>Password</td>
<td>Password for the database login.</td>
</tr>
<tr>
<td>Host</td>
<td>Name of the machine hosting the database server or FTP/SFTP host. For a FTP/SFTP connection, enter the host name or IP address.</td>
</tr>
<tr>
<td>Port</td>
<td>Network port number used to connect to FTP/SFTP connection. Default port is 21 for FTP and 22 for SFTP.</td>
</tr>
<tr>
<td>Local Directory</td>
<td>Directory on a local machine that stores the local file. The local machine must also run the Secure Agent used to run the corresponding task. Enter a local directory or use the Browse button to select a local directory.</td>
</tr>
<tr>
<td>Remote Directory</td>
<td>Directory on the FTP/SFTP host that stores the remote flat file. Depending on the FTP/SFTP server, you may have limited options to enter directories. For more information, see the FTP/SFTP server documentation.</td>
</tr>
<tr>
<td>Date Format</td>
<td>Date format for date fields in the flat file. Default date format is: MM/dd/yyyy HH:mm:ss</td>
</tr>
</tbody>
</table>
### FTP/SFTP Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Page</td>
<td>Code page compatible with the system where the source or target flat file resides. Select one of the following code pages:</td>
</tr>
<tr>
<td></td>
<td>- MS Windows Latin 1. Select for ISO 8859-1 Western European data.</td>
</tr>
<tr>
<td></td>
<td>- UTF-8. Select for Unicode data.</td>
</tr>
<tr>
<td>This is a Secure FTP</td>
<td>Indicates if the connection is secure or not secure. Select to create an SFTP connection.</td>
</tr>
<tr>
<td>Connection</td>
<td></td>
</tr>
</tbody>
</table>

For MS Access connections:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Secure Agent that Informatica Cloud uses to access the database in the local area network.</td>
</tr>
<tr>
<td>Data Source Name</td>
<td>System DSN name.</td>
</tr>
<tr>
<td>Code Page</td>
<td>The code page compatible with the MS Access database. Select one of the following code pages:</td>
</tr>
<tr>
<td></td>
<td>- MS Windows Latin 1. Select for ISO 8859-1 Western European data.</td>
</tr>
<tr>
<td></td>
<td>- UTF-8. Select for Unicode data.</td>
</tr>
</tbody>
</table>

For MySQL connections:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Secure Agent that Informatica Cloud uses to access the flat file or database in the local area network.</td>
</tr>
<tr>
<td>User Name</td>
<td>User name for the database login.</td>
</tr>
<tr>
<td>Password</td>
<td>Password for the database login.</td>
</tr>
<tr>
<td>Host</td>
<td>Name of the machine hosting the database server.</td>
</tr>
<tr>
<td>Port</td>
<td>Network port number used to connect to the database server. Default is 3306.</td>
</tr>
<tr>
<td>Database Name</td>
<td>Database name for the MySQL database target. Database name is case sensitive if the database is case sensitive. Maximum length is 64 characters. Valid characters are alphanumeric and underscore.</td>
</tr>
<tr>
<td>Code Page</td>
<td>The code page of the database server. Select one of the following code pages:</td>
</tr>
<tr>
<td></td>
<td>- MS Windows Latin 1. Select for ISO 8859-1 Western European data.</td>
</tr>
<tr>
<td></td>
<td>- UTF-8. Select for Unicode data.</td>
</tr>
</tbody>
</table>

For ODBC connections:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Secure Agent that Informatica Cloud uses to access the flat file or database in the local area network.</td>
</tr>
<tr>
<td>User Name</td>
<td>User name for the database login.</td>
</tr>
</tbody>
</table>
### ODBC Field

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
</tr>
<tr>
<td>Data Source Name</td>
</tr>
<tr>
<td>Schema</td>
</tr>
<tr>
<td>Code Page</td>
</tr>
</tbody>
</table>

#### Description

- **Password**: Password for the database login.
- **Data Source Name**: System DSN name.
- **Schema**: Schema used for the target connection. This field is not required for an ODBC connection, except if used to connect to an IBM DB2 database. Use uppercase letters when you specify the schema name for an Oracle database.
- **Code Page**: The code page of the database server or flat file defined in the connection. Select one of the following code pages:
  - MS Windows Latin 1. Select if the connection contains ISO 8859-1 Western European data.
  - UTF-8. Select if the connection contains Unicode data.

**For Oracle connections:**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
</tr>
<tr>
<td>User Name</td>
</tr>
<tr>
<td>Password</td>
</tr>
<tr>
<td>Host</td>
</tr>
<tr>
<td>Port</td>
</tr>
<tr>
<td>Service Name</td>
</tr>
<tr>
<td>Data Source Name</td>
</tr>
<tr>
<td>Schema</td>
</tr>
<tr>
<td>Code Page</td>
</tr>
</tbody>
</table>

#### Description

- **Agent**: Secure Agent that Informatica Cloud uses to access the database in the local area network.
- **User Name**: User name for the database login.
- **Password**: Password for the database login.
- **Host**: Name of the machine hosting the database server.
- **Port**: Network port number used to connect to the database server. Default is 1521.
- **Service Name**: Service name that uniquely identifies the Oracle database. If the connection fails, contact the database administrator.
- **Data Source Name**: System DSN name.
- **Schema**: Schema used for the target connection.
- **Code Page**: The code page of the database server. Select one of the following code pages:
  - MS Windows Latin 1. Select for ISO 8859-1 Western European data.
  - UTF-8. Select for Unicode data.

**For Salesforce connections:**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
</tr>
<tr>
<td>Password</td>
</tr>
<tr>
<td>Security Token</td>
</tr>
<tr>
<td>Service URL</td>
</tr>
</tbody>
</table>

#### Description

- **User Name**: User name for the Salesforce account.
- **Password**: Password for the Salesforce account.
- **Security Token**: Security token associated with the user name and password.
- **Service URL**: URL of the Salesforce service. Maximum length is 100 characters. Default service URL is: https://www.salesforce.com/services/Soap/u/19.0.
For SAP connections, configure the **SAP Connection Type**, and then enter the following IDoc Reader or IDoc Writer details:

<table>
<thead>
<tr>
<th>SAP IDoc Writer Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>SAP user name with authorization on S_DATASET, S_TABU_DIS, S_PROGRAM, and B_BTCH_JOB objects.</td>
</tr>
<tr>
<td>Password</td>
<td>Password for the SAP user name.</td>
</tr>
<tr>
<td>Connection String</td>
<td>Type A DEST entry in saprfc.ini.</td>
</tr>
<tr>
<td>Code Page</td>
<td>The code page compatible with the SAP target. Select one of the following code pages: - MS Windows Latin 1. Select for ISO 8859-1 Western European data. - UTF-8. Select for Unicode data.</td>
</tr>
<tr>
<td>Language Code</td>
<td>Language code that corresponds to the SAP language.</td>
</tr>
<tr>
<td>Client code</td>
<td>SAP client number.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SAP IDoc Reader Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Entry</td>
<td>Type R DEST entry in saprfc.ini. The Program ID for this destination entry must be the same as the Program ID for the logical system you defined in SAP to receive IDocs or consume business content data. For business content integration, set to INFACONTNT.</td>
</tr>
<tr>
<td>Code Page</td>
<td>The code page compatible with the SAP source. Select one of the following code pages: - MS Windows Latin 1. Select for ISO 8859-1 Western European data. - UTF-8. Select for Unicode data.</td>
</tr>
</tbody>
</table>

For SQL Server connections:

<table>
<thead>
<tr>
<th>SQL Server Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Secure Agent that Informatica Cloud uses to access the database in the local area network.</td>
</tr>
<tr>
<td>SQL Server Version</td>
<td>Microsoft SQL Server database version.</td>
</tr>
<tr>
<td>User Name</td>
<td>User name for the database login.</td>
</tr>
<tr>
<td>Password</td>
<td>Password for the database login.</td>
</tr>
<tr>
<td>Host</td>
<td>Name of the machine hosting the database server.</td>
</tr>
<tr>
<td>Port</td>
<td>Network port number used to connect to the database server. Default port number is 1433.</td>
</tr>
<tr>
<td>Instance Name</td>
<td>Instance name of the Microsoft SQL Server database.</td>
</tr>
<tr>
<td>Database Name</td>
<td>Database name for the Microsoft SQL Server target. Database name is case sensitive if the database is case sensitive. Maximum length is 100 characters. Database names can include alphanumeric and underscore characters.</td>
</tr>
</tbody>
</table>
### SQL Server Field

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema</td>
<td>Schema used for the target connection.</td>
</tr>
<tr>
<td>Code Page</td>
<td>The code page of the database server. Select one of the following code pages:</td>
</tr>
<tr>
<td></td>
<td>- MS Windows Latin 1. Select for ISO 8859-1 Western European data.</td>
</tr>
<tr>
<td></td>
<td>- UTF-8. Select for Unicode data.</td>
</tr>
</tbody>
</table>

For Web Service connections:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Secure Agent that Informatica Cloud uses to access the web service.</td>
</tr>
<tr>
<td>User Name</td>
<td>User name for the web service login. If the web service does not require a user name, leave this field empty.</td>
</tr>
<tr>
<td>Password</td>
<td>Password for the web service login. If the web service does not require a user name, leave this field empty.</td>
</tr>
<tr>
<td>Endpoint URL</td>
<td>Endpoint URL for the web service that you want to access. The WSDL file specifies this URL in the location element.</td>
</tr>
<tr>
<td>Domain</td>
<td>Domain for authentication.</td>
</tr>
<tr>
<td>Timeout</td>
<td>Number of seconds Informatica Cloud waits for a connection to the web service provider before it closes the connection and fails the session. Also, the number of seconds the Informatica Cloud waits for a SOAP response after sending a SOAP request before it fails the session. Default is 60 seconds.</td>
</tr>
<tr>
<td>Trust Certificates File</td>
<td>File containing the bundle of trusted certificates that Informatica Cloud uses when authenticating the SSL certificate of the web services provider. Default is ca-bundle.crt.</td>
</tr>
<tr>
<td>Certificate File</td>
<td>Client certificate that a web service provider uses when authenticating a client. You specify the client certificate file if the web service provider needs to authenticate Informatica Cloud.</td>
</tr>
<tr>
<td>Certificate File Password</td>
<td>Password for the client certificate. You specify the certificate file password if the web service provider needs to authenticate Informatica Cloud.</td>
</tr>
<tr>
<td>Certificate File Type</td>
<td>File type of the client certificate. You specify the certificate file type if the web service provider needs to authenticate the Integration Service. The file type can be either PEM or DER.</td>
</tr>
<tr>
<td>Private Key File</td>
<td>Private key file for the client certificate. You specify the private key file if the web service provider needs to authenticate Informatica Cloud.</td>
</tr>
<tr>
<td>Key Password</td>
<td>Password for the private key of the client certificate. You specify the key password if the web service provider needs to authenticate Informatica Cloud.</td>
</tr>
</tbody>
</table>
### Web Service Field

<table>
<thead>
<tr>
<th>Web Service Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key File Type</td>
<td>File type of the private key of the client certificate. You specify the key file type if the web service provider needs to authenticate Informatica Cloud. Informatica Cloud requires the PEM file type for SSL authentication.</td>
</tr>
</tbody>
</table>
| Authentication Type | Authentication type to use when the web service provider does not return an authentication type to Informatica Cloud:  
- Auto. The Integration Service attempts to determine the authentication type of the web service provider.  
- Basic. Based on a non-encrypted user name and password.  
- Digest. Based on an encrypted user name and password.  
- NTLM. Based on encrypted user name, password, and domain.  
Default is Auto. |

4. To test the connection, click **Test**.  
   If a database connection fails, contact the database administrator.  
5. Click **OK** to save the connection.

### Viewing Connection Details

You can view details about a connection, such as the login name of the creator, the date it was created, and the connection properties.

- Click **Configuration > Connections**, and then click the connection name.  
  If you want to edit connection details, click **Edit**.

### Copying a Connection

You can copy a connection. You might copy an existing connection to use some of the same connection attributes for a different connection object. For example, you might have two flat file connections that have the same connection attributes, except for a different directory where the flat file is stored.

1. Click **Configuration > Connections**.  
2. Click the **Make a Copy** icon next to the connection name.

**RELATED TOPICS:**
- “Rules and Guidelines for Copying Objects” on page 156

### Testing a Connection

You can test the connection after you enter the connection details.

1. To test a new connection, click **Configuration > Connections**, and then click **New**.  
   To test an existing connection, click **Configuration > Connections**, click the connection name.  
2. Click **Test**.
**Deleting a Connection**

Delete a connection if you no longer need to connect to the corresponding source or target. You cannot delete a connection if it is used for a custom source or in a task. Before you delete the connection, edit the associated custom source or task to use a different connection.

1. Click **Configuration > Connections**.
2. On the Connections page, click the **Delete** icon next to the connection name.

**Schedules**

Create schedules to run tasks at specified times or on regular intervals. After you create a schedule, you can associate it with one or more tasks.

You can perform the following tasks for schedules:

- Create a schedule.
- Edit a schedule.
- View schedule details.
- Delete a schedule.

**Repeat Frequency**

The repeat frequency determines how often tasks run. The following table describes the repeat frequency options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Does not repeat</strong></td>
<td>Tasks run as scheduled and do not repeat.</td>
</tr>
<tr>
<td><strong>Every N minutes</strong></td>
<td>Tasks run on an interval based on a specified number of minutes. You can configure the following options:</td>
</tr>
<tr>
<td></td>
<td>- Repeat frequency. Select a frequency in minutes. Options are 5, 10, 15, 20, 30, 45.</td>
</tr>
<tr>
<td></td>
<td>- Days. Days of the week when you want tasks to run. You can select one or more days of the week.</td>
</tr>
<tr>
<td></td>
<td>- Time range. Hours of the day when you want tasks to start. Select All Day or configure a time range. You can configure a time range between 00:00-23:55.</td>
</tr>
<tr>
<td></td>
<td>- Repeat option. The range of days when you want tasks to run. You can select Repeat Indefinitely or configure an end date and time.</td>
</tr>
<tr>
<td><strong>Hourly</strong></td>
<td>Tasks run on an hourly interval based on the start time of the schedule.</td>
</tr>
<tr>
<td></td>
<td>You can configure the following options:</td>
</tr>
<tr>
<td></td>
<td>- Repeat frequency. Select a frequency in hours. Options are 1, 2, 3, 4, 6, 8, 12.</td>
</tr>
<tr>
<td></td>
<td>- Days. Days of the week when you want tasks to run. You can select one or more days of the week.</td>
</tr>
<tr>
<td></td>
<td>- Time range. Hours of the day when you want tasks to start. Select All Day or configure a time range. You can configure a time range between 00:00-23:55.</td>
</tr>
<tr>
<td></td>
<td>- Repeat option. The range of days when you want tasks to run. You can select Repeat Indefinitely or configure an end date and time.</td>
</tr>
<tr>
<td><strong>Daily</strong></td>
<td>Tasks run daily at the start time configured for the schedule.</td>
</tr>
<tr>
<td></td>
<td>You can configure the following options:</td>
</tr>
<tr>
<td></td>
<td>- Repeat frequency. The frequency at which you want tasks to run. Select Every Day or Every Weekday.</td>
</tr>
<tr>
<td></td>
<td>- Repeat option. The range of days when you want tasks to run. You can select Repeat Indefinitely or configure an end date and time.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Weekly | Tasks run on a weekly interval based on the start time of the schedule. You can configure the following options:  
- Days. Days of the week when you want tasks to run. You can select one or more days of the week.  
- Repeat option. The range of days when you want tasks to run. You can select Repeat Indefinitely or configure an end date and time.  
If you do not specify a day, the schedule runs regularly on the same day of the week as the start date. |
| Monthly | Tasks run on a monthly interval based on the start time of the schedule. You can configure the following options:  
- Day. Day of the month when you want tasks to run. You can configure one of the following options:  
  - Select the exact date of the month, between 1-28. If you want the task to run on days later in the month, use the &lt;day of the week&gt; option.  
  - Select the &lt;day of the week&gt;. Options for &lt;day of the week&gt; include Day, and Sunday-Saturday.  
  Tip: With the Day option, you can configure tasks to run on the First Day or the Last Day of the month.  
- Repeat option. The range of days when you want tasks to run. You can select Repeat Indefinitely or configure an end date and time. |

**Time Zones and Schedules**

Informatica Cloud stores time in Coordinated Universal Time (UTC). When you log in to Informatica Cloud, Informatica Cloud converts the time and displays it in the time zone associated with your user account.

**Note:** Informatica Cloud adjusts times accordingly when Daylight Saving Time is in effect.

**Daylight Savings Time Changes and Schedules**

Informatica Cloud applies Daylight Savings Time changes to tasks that run every day or every week.

**Note:** Informatica Cloud does not support Daylight Savings Time changes for schedules that run at other frequencies, such as hourly or monthly.

Daylight Savings Time does not trigger additional runs for tasks that are scheduled to run between 1:00 a.m. - 1:59 a.m and run everyday or every week. For example, a task is scheduled to run everyday at 1:30 a.m. When the clock changes from 2 a.m. to 1 a.m. during Daylight Savings Time, the task does not run again at 1:30 a.m.

On the other hand, if a task is scheduled to run everyday or every week between 2:00 a.m. and 2:59 a.m., it does not run the day that the clock changes forward from 2:00 a.m. to 3:00 a.m.

**Tip:** To ensure Informatica Cloud does not skip any scheduled runs near the 2 a.m. time change, do not schedule jobs to run between 12:59 a.m. - 3:01 a.m.

**Configuring a Schedule**

You can configure a schedule while configuring a task or on the Schedules page. You can configure a schedule to run tasks once or at a specific repeat frequency, indefinitely or until a specified end time.

When you create schedules for tasks, you specify the date and time. You can configure a schedule to run tasks throughout the day between 12:00 a.m. and 11:55 p.m.

The date and time for a schedule are in the time zone associated with the user who creates the schedule. When you create schedules for tasks that write to database targets, consider the time zone associated with your user account and the time zone of the target database.
For example, you log in as a user in the Eastern Standard time zone (UTC-5). The server where the target database resides is located in the Pacific Standard time zone (UTC-8). You want to create a schedule that replicates data to the target database each night at 11:55 p.m. in the server time zone. You configure the schedule to replicate data each morning at 2:55 a.m. EST.

Informatica Cloud might add a small schedule offset to the start time, end time, and all other time configuration for schedules. As a result, scheduled tasks might start later than expected. For example, you configure a schedule to run hourly until noon, and the schedule offset for your organization is 10 seconds. Informatica Cloud extends the end time for the schedule to 12:00:10 p.m., and the last hourly task run starts at 12:00:10 p.m.

To see the schedule offset for your organization, check the Schedule Offset organization property.

To configure a schedule:

1. To create a schedule, click **Configuration > Schedules**, and then click **New**.
   
   To edit a schedule, click **Configuration > Schedules**, click the schedule name in the list of schedules, and then click **Edit**.

2. On the New Schedule page or Edit Schedule page, configure the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule Name</td>
<td>Name of the schedule. Each schedule name must be unique within the organization. Schedule names can contain alphanumeric characters, spaces, and the following special characters: _ - + . Schedule names are not case sensitive.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the schedule. Maximum length is 255 characters.</td>
</tr>
<tr>
<td>Starts</td>
<td>Date and time when the schedule starts. The date format is MM/DD/YYYY. Time appears in the 24-hour format. Click the Calendar button to select the start date. The start date and time can affect the repeat frequency for tasks that repeat at regular intervals. For example, if the start date is November 10 and the repeat frequency is monthly, the tasks run on the tenth day of each month. If the start time is 3:10 and the repeat frequency is hourly, the tasks run every hour at 10 minutes past the hour. Default is the current date, current time, and time zone of the user who is creating the schedule.</td>
</tr>
<tr>
<td>Repeats</td>
<td>Repeat frequency for the schedule. Select one of the following options: Does Not Repeat, Every N Minutes, Hourly, Daily, Weekly, Monthly. Default is Does Not Repeat.</td>
</tr>
</tbody>
</table>
3. Complete the Repeat Frequency Options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does not repeat</td>
<td>Tasks run as scheduled and do not repeat.</td>
</tr>
<tr>
<td>Every N minutes</td>
<td>Tasks run on an interval based on a specified number of minutes. You can configure the following options:</td>
</tr>
<tr>
<td></td>
<td>- Repeat frequency. Select a frequency in minutes. Options are 5, 10, 15, 20, 30, 45.</td>
</tr>
<tr>
<td></td>
<td>- Days. Days of the week when you want tasks to run. You can select one or more days of the week.</td>
</tr>
<tr>
<td></td>
<td>- Time range. Hours of the day when you want tasks to start. Select All Day or configure a time range.</td>
</tr>
<tr>
<td></td>
<td>- You can configure a time range between 00:00-23:55.</td>
</tr>
<tr>
<td></td>
<td>- Repeat option. The range of days when you want tasks to run. You can select Repeat Indefinitely or configure an end date and time.</td>
</tr>
<tr>
<td>Hourly</td>
<td>Tasks run on an hourly interval based on the start time of the schedule. You can configure the following options:</td>
</tr>
<tr>
<td></td>
<td>- Repeat frequency. Select a frequency in hours. Options are 1, 2, 3, 4, 6, 8, 12.</td>
</tr>
<tr>
<td></td>
<td>- Days. Days of the week when you want tasks to run. You can select one or more days of the week.</td>
</tr>
<tr>
<td></td>
<td>- Time range. Hours of the day when you want tasks to start. Select All Day or configure a time range.</td>
</tr>
<tr>
<td></td>
<td>- You can configure a time range between 00:00-23:55.</td>
</tr>
<tr>
<td></td>
<td>- Repeat option. The range of days when you want tasks to run. You can select Repeat Indefinitely or configure an end date and time.</td>
</tr>
<tr>
<td>Daily</td>
<td>Tasks run daily at the start time configured for the schedule. You can configure the following options:</td>
</tr>
<tr>
<td></td>
<td>- Repeat frequency. The frequency at which you want tasks to run. Select Every Day or Every Weekday.</td>
</tr>
<tr>
<td></td>
<td>- Repeat option. The range of days when you want tasks to run. You can select Repeat Indefinitely or configure an end date and time.</td>
</tr>
<tr>
<td>Weekly</td>
<td>Tasks run on a weekly interval based on the start time of the schedule. You can configure the following options:</td>
</tr>
<tr>
<td></td>
<td>- Days. Days of the week when you want tasks to run. You can select one or more days of the week.</td>
</tr>
<tr>
<td></td>
<td>- Repeat option. The range of days when you want tasks to run. You can select Repeat Indefinitely or configure an end date and time.</td>
</tr>
<tr>
<td></td>
<td>If you do not specify a day, the schedule runs regularly on the same day of the week as the start date.</td>
</tr>
<tr>
<td>Monthly</td>
<td>Tasks run on a monthly interval based on the start time of the schedule. You can configure the following options:</td>
</tr>
<tr>
<td></td>
<td>- Day. Day of the month when you want tasks to run. You can configure one of the following options:</td>
</tr>
<tr>
<td></td>
<td>- Select the exact date of the month, between 1-28. If you want the task to run on days later in the month, use the &lt;n&gt; &lt;day of the week&gt; option.</td>
</tr>
<tr>
<td></td>
<td>- Select the &lt;n&gt; &lt;day of the week&gt; option. Options for &lt;n&gt; include First, Second, Third, Fourth, and Last. Options for &lt;day of the week&gt; includes Day, and Sunday-Saturday.</td>
</tr>
<tr>
<td></td>
<td>Tip: With the Day option, you can configure tasks to run on the First Day or the Last Day of the month.</td>
</tr>
<tr>
<td></td>
<td>- Repeat option. The range of days when you want tasks to run. You can select Repeat Indefinitely or configure an end date and time.</td>
</tr>
</tbody>
</table>

4. Click OK.

**Viewing Schedule Details**

You can view details about a schedule, such as the start and end dates and repeat frequency. When you view the details, click Edit to modify the schedule.

- Click **Configuration > Schedules**, and then click the schedule name.
Deleting a Schedule

Delete a schedule if you no longer need to run tasks on the schedule. If a task uses the schedule, you must remove the schedule from the task before you can delete the schedule.

1. Click Configuration > Schedules.
2. On the Schedules page, click the Delete icon next to the schedule name.

Task Flows

You can create a task flow to group multiple tasks. You can run the task flow immediately or on a schedule. The task flow runs tasks serially, in the specified order.

You can also configure email notification options for a task flow.

Configuring a Task Flow

You can create a task flow to run multiple tasks in a specific order. You cannot add a task more than one time to a task flow. You can add any type of task, except data loader tasks, to a task flow.

You can edit a task flow. If you add a task to a task flow that is currently running, Informatica Cloud does not run the new task until the next time the task flow runs.

To configure a task flow:

1. To create a task flow, click Configuration > Task Flows, and then click New.
   To edit a task flow, click Configuration > Task Flows. Click the task flow name, and then click Edit.
2. On the Task Flows page or Edit Task Flows page, enter the following task flow details:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Flow Name</td>
<td>Name of the task flow. The names of the task flows must be unique within the organization. Names can contain alphanumeric characters, spaces, and the following special characters: <code>_ . + -</code> Names are not case sensitive.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the task flow. Maximum length is 255 characters.</td>
</tr>
</tbody>
</table>
| Schedule    | Determines how the task runs:  
   - To run the task manually, click Do Not Run This Task on a Schedule.  
   - To associate the task with a schedule, click Run This Task on Schedule, and select a schedule.  
   To create a schedule, click New.                                      |

3. To add a task to the task flow, click Add Task.
4. In the Add Task dialog box, select the task type, and then select the task.
5. Click OK.
6. Repeat steps 3 to 5 to add additional tasks.
7. Enter the sequence numbers for all tasks.
   The sequence numbers determine the order in which the tasks run.
8. If you want the task flow to stop if a task fails, click Stop on Error for the task.
The applicable service stops running all remaining tasks if any of the selected tasks fail. If you change this option while the task flow is running, Informatica Cloud does not apply the change until the next time the task flow runs.

9. To delete a task from the task flow, click the **Delete** icon next to the task.

10. Configure email notification options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Default Email Notification Options for my Organization</td>
<td>Use the email notification options configured for the organization.</td>
</tr>
</tbody>
</table>
| Use Custom Email Notification Options for this Task | Use the email notification options configured for the task. You can send email to different addresses based on the status of the task:  
- Error Email Notification. Sends email to the listed addresses when the task fails to complete.  
- Warning Email Notification. Sends email to the listed addresses when the task completes with errors.  
- Success Email Notification. Sends email to the listed addresses when the task completes without errors.  
Use commas to separate a list of email addresses.  
When you select this option, email notification options configured for the organization are not used. |

11. Click **OK**.

**Viewing Task Flow Details**

You can view details about a task flow, such as the name, description, schedule, and list of tasks included in the task flow. When you view the details, click **Edit** to modify the task flow.

To view task flow details:

> Click **Configuration > Task Flows**, and then click the task flow name.

**Running a Task Flow**

You can run a task flow in the following ways:

- **Manually**  
  Runs the task flow at any time. When you run a task flow manually, the task flow starts immediately.

- **On a schedule**  
  Configure a task flow to run on a schedule. When you configure a task flow to run on a schedule, include a repeat frequency to run the task flow on regular intervals.

**Running a Task Flow Manually**

Run a task flow manually to immediately run the batch of tasks. You might want to run a task flow manually for the following reasons:

- To verify that the task flow and tasks are configured properly.
- To run the batch of tasks occasionally. You may not want to run the batch of tasks at regular intervals.

When you start the task flow manually, the activity monitor appears and shows you details about each task in the task flow.
To run a task flow manually:
1. Click Configuration > Task Flows.
2. On the Task Flows page, click the Run Now icon for the task flow that you want to run.

Running a Task Flow on a Schedule
Use a schedule to run the task flow at a specified time or at regular intervals. Edit the task flow to change the schedule associated with it.
1. Click Configuration > Task Flows.
2. Select a schedule in the Schedule field, or click New to create a schedule.

Deleting a Task Flow
Delete a task flow if you no longer need to run the tasks as a group. You can delete a task flow at any time. Before you delete a task flow, verify that no users in the organization plan to use it. You cannot retrieve a task flow after you delete it.
1. Click Configuration > Task Flows.
2. On the Task Flows page, click the Delete icon next to the name of the task flow.

Plug-ins
An Informatica Cloud plug-in is transformation logic that you can create and use to transform source data before it is loaded into the target. You create the logic in a mapplet in PowerCenter, and then import the logic into Informatica Cloud. You can use one plug-in in each data synchronization task.

To work with a plug-in, your organization must have the Plug-in license.

You can perform the following tasks for plug-ins:
- Create, edit, or delete a plug-in.
- View a plug-in.
- Add a plug-in to a data synchronization task.
- Remove a plug-in from a data synchronization task.

For information about obtaining the Plug-in license, contact Informatica Global Customer Support.

Active and Passive Plug-ins
A plug-in can be active or inactive. An active plug-in is derived from a mapplet that contains at least one active PowerCenter transformation. An inactive plug-in is derived from a mapplet that contains all passive PowerCenter transformations.

When you create the plug-in, you must specify whether the plug-in is active or inactive. An active plug-in may return a different number of rows that were input. For example, a plug-in can aggregate five source rows and return one row. As a result, when you add an active plug-in to a data synchronization task, you cannot map the source fields to the target fields. You must map the source fields to the plug-in, and then from the plug-in to the target.
Stored Procedures in Plug-ins

When you use a Stored Procedure transformation in a workflow for a plug-in, the stored procedure must include exception handling. Exception handling can be as complex as necessary. Or, you can use the following simple example of exception handling code:

```sql
Exception
when NO_DATA_FOUND
then NULL;
END;
```

For example, you have the following stored procedure in a PowerCenter workflow:

```sql
CREATE OR REPLACE PROCEDURE SP_GETSAL_WITH_EXCEPTION (EMP_ID NUMBER, EMP_NAME OUT VARCHAR, SAL OUT NUMBER)
AS
BEGIN
    SELECT EMPNAME INTO EMP_NAME FROM EMPLOYEE WHERE EMPID=EMP_ID;
    SELECT SALARY INTO SAL FROM EMPLOYEE WHERE EMPID=EMP_ID;
END;
```

Before you export the workflow, add exception handling as follows:

```sql
CREATE OR REPLACE PROCEDURE SP_GETSAL_WITH_EXCEPTION (EMP_ID NUMBER, EMP_NAME OUT VARCHAR, SAL OUT NUMBER)
AS
BEGIN
    SELECT EMPNAME INTO EMP_NAME FROM EMPLOYEE WHERE EMPID=EMP_ID;
    SELECT SALARY INTO SAL FROM EMPLOYEE WHERE EMPID=EMP_ID;
    Exception
    when NO_DATA_FOUND
    then NULL;
    END;
```

Rules and Guidelines for the PowerCenter XML File

Use the following rules and guidelines for the PowerCenter XML file used for a plug-in:

- The XML file must contain exactly one workflow, session, mapping, and mapplet.
- The PowerCenter workflow must contain a valid session and mapping.
- The PowerCenter session can use any type of connections. However, it is easier if the session uses flat file connections for the source and target.
- You do not have to map all source and target fields in the PowerCenter mapping.
- The PowerCenter mapping can contain the following supported transformations:
  - Aggregator transformation
  - Expression transformation
  - Filter transformation
  - HTTP transformation
  - Lookup transformation
  - Salesforce Lookup transformation (multiple matches returns a single match)
  - Salesforce Picklist transformation
  - Salesforce Merge transformation
  - Sorter transformation
  - Stored procedure transformation with exception handling
  - Transaction Control transformation
  - Web Services consumer transformation
  - XML Generator transformation with flat file or database sources
- XML Parser transformation with flat file or database sources
  
  • The underlying PowerCenter mapplet cannot contain multiple Input transformations.
  
  • If the PowerCenter mapping or mapplet contains a Stored Procedure transformation, the stored procedure must include exception handling.

Rules and Guidelines for Using Plug-ins in Tasks

Use the following rules and guidelines for using plug-ins in a data synchronization task:

• You can add one plug-in to each data synchronization task.
• You can add expressions and lookups between the source and a plug-in.
• You can add expressions between a plug-in and the target.
• When you add a plug-in to a data synchronization task, the Data Synchronization Task wizard removes existing field mappings, except when you add a passive plug-in between the source and target.
• When you use an active plug-in with a data synchronization task that includes a custom source, the Data Synchronization service ignores the configured target option for the task and tries to insert data to the target.
• You can click the Edit Plug-in button in step 5 of the Data Synchronization Task wizard to edit the connection.

Configuring a Plug-in

You can create plug-in logic in PowerCenter 8.6.1, and then import the logic into Informatica Cloud.

Before you can import the logic as a plug-in, perform the following prerequisite tasks:

1. Create the plug-in logic in a mapplet in PowerCenter.
2. Validate the mapplet in PowerCenter.
3. Create a mapping, session, and workflow for the mapplet.
4. Validate the mapping, session, and workflow in PowerCenter.
5. Export the workflow from PowerCenter to an XML file.

After you complete the prerequisite tasks, import the XML file into Informatica Cloud. When you import the XML file, Informatica Cloud creates a plug-in based on the mapplet definition in the XML file. Informatica Cloud retains PowerCenter session-level overrides to the mapping.

To configure a plug-in:

1. Click Configuration > Plug-ins.
2. To import the XML file into Informatica Cloud, click New.
   To edit the plug-in, click the Edit icon next to the plug-in.
3. On the New Plug-in or Edit Plug-in page, configure the following details:

<table>
<thead>
<tr>
<th>Detail</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plug-in Name</td>
<td>Name of the plug-in.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the plug-in.</td>
</tr>
<tr>
<td>Active</td>
<td>Indicates the plug-in is active.</td>
</tr>
</tbody>
</table>

4. To upload the Mapplet XML file, click Upload.
5. In the Upload Mapplet XML File dialog box, click **Browse**.
6. Browse to the appropriate location and select the XML file.
7. Click **OK**.
   The Mapplet XML File Details area shows the connections, input fields, and output fields.
8. Click **OK**.
   The plug-in appears in the Plug-ins page.

**Viewing a Plug-in**

To view a plug-in:
1. Click **Configuration > Plug-ins**.
2. The Plug-ins page lists the plug-ins.
3. Click the name of a plug-in to view the details about the plug-in.

**Deleting a Plug-in**

You cannot delete a plug-in that is used in a data synchronization task. Remove the plug-in from the task before you delete the plug-in.
1. Click **Configuration > Plug-ins**.
2. The Plug-ins page lists the plug-ins.
3. Click the **Delete** icon next to the plug-in.
   A warning message appears.
4. To delete the plug-in, click **OK**.

**Custom Sources**

A custom source is a source object that you can create from one or more database tables. Create a custom source when you want to use a database source that you cannot configure using the single- or multiple-object source options. For example, you might create a custom source to include source filters, or to perform a complicated join of multiple tables. You can add one custom source in each data synchronization task.

To use custom sources, your organization must have the Custom Sources license.

You can perform the following tasks with custom sources:
- Create, edit, or delete a custom source.
- View a custom source.
- Copy a custom source.
- Add a custom source to a data synchronization task.
- Remove a custom source from a data synchronization task.
- Run a task that includes a custom source.

**Note**: You cannot migrate a data synchronization task that contains a custom source.

For information about obtaining the Custom Source license, contact Informatica Global Customer Support.
Configuring a Custom Source

When you create a custom source, enter an SQL SELECT statement to select the source columns you want to use. Use an SQL statement that is valid for the source database.

Informatica Cloud uses the SQL statement to retrieve source column information. You can edit datatype, precision, or scale of each column before you save the custom source.

For example, you might create a custom source based on a TRANSACTIONS table that includes transactions from 2010 with the following SQL statement:

```
SELECT TRANSACTION_ID, TRANSACTION_TOTAL, TRANSACTION_TIMESTAMP from dbo.TRANSACTIONS WHERE TRANSACTION_TIMESTAMP>'0:0:0:0 01/01/2010'
```

Informatica Cloud ensures that custom source column names are unique. If an SQL statement returns a duplicate column name, Informatica Cloud adds a number to the duplicate column name as follows:

```
<column_name><number>
```

**Tip:** Test the SQL statement you want to use on the source database before you create a custom source.

Rules and Guidelines for Configuring a Custom Source

Use the following rules and guidelines to configure a custom source:

- When you edit a column datatype, precision, or scale, error rows are not written to the error log file at runtime.
- Specific error messages for invalid SQL statements do not display. For more information about an invalid SQL statement, test the statement on the source database.
- In the SQL statement:
  - Use a SELECT statement.
  - Do not use an asterisk (*) to select all columns of a table. List the columns that you want to select.
  - Do not use conversion functions, such as TO_CHAR or TO_DATE.
  - Do not use the COUNT function.
  - For an Oracle source database, omit the final semicolon (;) from the statement.
  - For a MySQL source database, do not use a full outer join.
  - For a Microsoft Access source database, do not use the DATEADD function.
  - For a MySQL source database, Float columns import as Real(7) and Double columns import as Decimal(22, 21). You can edit the imported column information before you save the custom source.

Rules and Guidelines for Using Custom Sources in Tasks

Use the following rules and guidelines for using a custom source in a data synchronization task:

- You can add one custom source to each data synchronization task.
- You cannot delete a custom source that is used in a data synchronization task.
- You cannot migrate a data synchronization task that includes a custom source.
- Error rows are not written to the error log file at runtime if column information for the custom source was edited.
- In the activity log details for the task, the success and error row counts for sources are not accurate.
- When you use an active plug-in with a data synchronization task that includes a custom source, the Data Synchronization service ignores the configured target option for the task and tries to insert data to the target.
Steps to Create a Custom Source

1. Click **Configuration > Custom Sources**.
2. On the Custom Sources page, click **New**.
3. On the New Custom Source page, enter the following information:

<table>
<thead>
<tr>
<th>Custom Source Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom Source Name</td>
<td>Name of the custom source.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the custom source.</td>
</tr>
<tr>
<td>Database Type</td>
<td>Source database type. Select one of the following database types:</td>
</tr>
<tr>
<td>- Oracle</td>
<td></td>
</tr>
<tr>
<td>- SQL Server</td>
<td></td>
</tr>
<tr>
<td>- MySQL</td>
<td></td>
</tr>
<tr>
<td>- ODBC</td>
<td></td>
</tr>
<tr>
<td>- MS Access</td>
<td></td>
</tr>
<tr>
<td>SQL Query</td>
<td>Enter a valid SQL statement to select the source columns you want to use.</td>
</tr>
</tbody>
</table>

4. Click **Get Columns**.
5. In the Get Columns dialog box, select a connection.
   The Custom Source Column Detail table displays the source columns selected in the SQL statement. In the table, the Info column displays a summary of information about the column. Type displays the datatype for the column.
6. In the Custom Source Column Detail table, you can edit the datatype, precision, or scale.
   If you edit these values, Informatica Cloud does not write error rows into the error log file.
7. To change all source column information, enter another SQL statement and click **Get Columns**.
   Informatica Cloud deletes all information in the Custom Source Column Detail table, and then retrieves column information based on the latest SQL statement.
   You can edit the datatype, precision, or scale of the columns.
8. Click **OK**.

Viewing a Custom Source

To view a custom source.

1. Click **Configuration > Custom Sources**.
   The Custom Sources page lists all custom sources.
2. Click the name of a custom source that you want to view.

Copying a Custom Source

You can copy a custom source. You might copy a custom source to use it on a different database, or to create another custom source based on a similar SQL statement.

1. Click **Configuration > Custom Sources**.
   The Custom Sources page lists all custom sources.
2. Click the **Make a Copy** icon next to the custom source that you want to delete.  
   A copy of the custom source appears in the list, named `<custom_source>_<version_number>`.

**Deleting a Custom Source**

You cannot delete a custom source that is used in a data synchronization task. Before you delete the custom source, delete the task or remove the custom source from the task.

1. Click **Configuration > Custom Source**.
2. The Custom Sources page lists all custom sources.
3. Click the **Delete** icon next to the custom source.  
   A warming message displays.
4. Click **OK**.
CHAPTER 4

Data Assessment Service

This chapter includes the following topics:
- Data Assessment Service Overview, 67
- Understanding Quality Metrics, 68
- Understanding Scorecards, 70
- Rules and Guidelines for Data Assessment Sources, 72
- Configuring a Data Assessment Task, 72
- Customizing Weights, 76
- Customizing Field Mappings, 77
- Viewing Data Assessment Task Details, 78
- Copying a Data Assessment Task, 78
- Running a Data Assessment Task, 78
- Stopping a Data Assessment Task, 79
- Monitoring a Data Assessment Task, 79
- Viewing the Scorecard for a Data Assessment Task, 79
- Analyzing Data Assessment Results, 79
- Deleting a Data Assessment Task, 80

Data Assessment Service Overview

The Data Assessment service allows you to evaluate the quality of your Salesforce data. It generates graphical dashboards that measure field completeness, field conformance, record duplication, and address validity for each Salesforce object. You can run data assessment tasks on an on-going basis to show trends in the data quality.

A data assessment task is a process that you configure to analyze the quality of a data set. You can create one or more data assessment tasks for each Salesforce object. When you create a data assessment task, you specify the Salesforce object and the quality metrics to include in the plan. Each data quality metric provides a particular type of data quality analysis.

You can configure filters on the data and thresholds, which determine unacceptable, acceptable, and good data. Optionally, assign a data assessment task to a schedule to run it at a specified time or at regular intervals.

When you run the data assessment task, the Data Assessment service runs the quality check for each data quality metric included in the plan and publishes the results in a scorecard. You can view the scorecard of a data assessment task on the dashboard and on the Data Assessment Results page. You can also export and analyze the underlying Salesforce records that determine the scorecard results.
Understanding Quality Metrics

You can add one or more of the following quality metrics to a data assessment task:

- Address validation
- Completeness
- Conformance
- Duplicates

Address Validation Quality Metric

Use the Address Validation quality metric to validate United States and Canada address information for a Salesforce object. The metric determines the percentage of address-related fields for a Salesforce object that have valid address information.

When you create the data assessment task, indicate if you want the data assessment task to perform address validation. If you include address validation, select the type of addresses to be validated. By default, the plan validates shipping and billing addresses. You can validate addresses in custom Salesforce fields by mapping them to the billing or shipping address plan fields.

When validating addresses, the Data Assessment service compares the address information in each field selected for address validation against address reference datasets provided with Informatica Cloud. The Data Assessment service assumes the address value in the field is not valid if the value does not match a value in the address reference. The validation check is not case sensitive. The Data Assessment service counts null values in address fields as not valid.

Completeness Quality Metric

Use the Completeness quality metric to verify that each field does not contain blank or null values. The metric determines the percentage of fields for a Salesforce object that do not have blank or null values. The Data Assessment service can validate completeness of all types of fields for a Salesforce object.

When you create the data assessment task, indicate if you want the data assessment task to perform a completeness check. If you include the Completeness quality metric, you can select the fields to check for completeness. For example, you omit a field that is rarely populated in Salesforce and that is not important to your organization.

Conformance Quality Metric

Use the Conformance quality metric to determine the percentage of fields for a Salesforce object that conform to a predefined format. Conformance applies to particular types of fields. When you create the data assessment task, indicate if you want the data assessment task to perform a conformance check. If you include the Conformance quality metric, you can select the fields to be verified for conformance.

Duplicates Quality Metric

Use the Duplicates quality metric to determine whether there are duplicate records for a Salesforce object. The metric determines the percentage of duplicate records for a given Salesforce object. The Data Assessment service determines whether records in the same group are duplicates based on a patented, fuzzy matching algorithm, field weights, and a threshold. The matching algorithm is not case sensitive.

Before comparing records, the Data Assessment service groups the records of each Salesforce object based on a field. The Data Assessment service then compares records within each group based on the matching algorithm,
field weights, and threshold. If two records are in different groups, the Data Assessment service assumes the records are not duplicates of each other.

The following table lists the field used to group records for each type of Salesforce object:

<table>
<thead>
<tr>
<th>Salesforce Object</th>
<th>Field to Group By</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account</td>
<td>BillingPostalCode</td>
</tr>
<tr>
<td>Contact</td>
<td>MailingPostalCode</td>
</tr>
<tr>
<td>Lead</td>
<td>PostalCode</td>
</tr>
<tr>
<td>Opportunity</td>
<td>Probability</td>
</tr>
</tbody>
</table>

When you create the data assessment task, specify the fields of a record to compare to determine whether two records are duplicates. You can configure the data assessment task to compare all or some of the fields in a record.

Customize the weights assigned to fields to specify how significant the fields are in determining duplicate records. You can also specify the threshold at which records are considered duplicates. The Data Assessment service uses a fuzzy matching algorithm to determine duplicates.

When comparing two records, the Data Assessment service compares the values for each field that is included in the Duplicates quality metric. The Data Assessment service assigns a score to each field based on how closely the values match. Each score ranges from 0 to 1, where 0 indicates no match and 1 indicates an exact match. The Data Assessment service adds the product of each score and the corresponding weight to determine a matching score for the two records. If the matching score exceeds or is equal to the threshold, the Data Assessment service considers the rows to be duplicates. The Data Assessment service provides the percentage of all records that are duplicates in the scorecard for the data assessment task.

**Note:** The Data Assessment service uses a matching algorithm that is not case sensitive.

**Example of Records in Different Groups**

Two records for the Salesforce Account object have the same values for all fields, except the BillingPostalCode field. Regardless of field weights, threshold, and matching algorithm, the Data Assessment service organizes the records in different groups. The Data Assessment service assumes the records are not duplicates because they are in different groups.

**Example of Records in Same Group**

You create a data assessment task to determine the percentage of duplicate records for the Salesforce Account object.

You configure the following custom weights for the data assessment task fields for the Salesforce Account object:

<table>
<thead>
<tr>
<th>Field</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountNumber</td>
<td>50</td>
</tr>
<tr>
<td>Name</td>
<td>40</td>
</tr>
<tr>
<td>BillingPostalCode</td>
<td>10</td>
</tr>
</tbody>
</table>

You set the Threshold Value field to 65.
The Data Assessment service determines that the two records are in the same group because they have the same BillingPostalCode. Next, the Data Assessment service compares the values for the AccountNumber, Name, and BillingPostalCode fields for two records.

The Data Assessment service assigns the following scores to the fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccountNumber</td>
<td>1</td>
<td>Account numbers are the same.</td>
</tr>
<tr>
<td>Name</td>
<td>0.3</td>
<td>Account names are similar.</td>
</tr>
<tr>
<td>BillingPostalCode</td>
<td>1</td>
<td>Billing postal codes are the same.</td>
</tr>
</tbody>
</table>

The Data Assessment service uses the following calculation to determine the matching score for the two records:

\[ 1 \times 50 + 0.3 \times 40 + 1 \times 10 = 72 \]

The Data Assessment service compares the matching score, 72, against the Threshold Value, 65. The Data Assessment service considers the two records duplicates because the matching score for the two records is higher than or equal to the threshold.

**Note:** You can set different weights or a different threshold and have a different outcome with the same two records.

---

**Understanding Scorecards**

The scorecard provides the following types of details:

- Top issues
- Field scores
- Individual scores
- Overall score
- Trend chart

**Top Issues**

The Top Issues area of the scorecard shows the top issues based on the results of all quality metrics included in the data assessment task. For example, if you included the Duplicates and Address Validation quality metrics in the data assessment task, this area provides the top issues for both quality metrics.

The Top Issues area rounds percentages up to the nearest percent. For example, if 6 out of 100,000 records are incomplete, the Top Issues area shows 1% incomplete.

Use the Top Issues area to identify higher-priority data quality issues with the data for the associated Salesforce object.

**Field Scores**

The Field Scores area of the scorecard shows field scores for Completeness and Conformance quality metrics. The Field Scores area shows the number of records that have complete or conforming data for each field included in the quality metric.
For example, a scorecard shows the following field scores:

<table>
<thead>
<tr>
<th>Field</th>
<th>Complete</th>
<th>Conformant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Billing Street</td>
<td>19745</td>
<td></td>
</tr>
<tr>
<td>Created Date</td>
<td>20000</td>
<td></td>
</tr>
<tr>
<td>Fax</td>
<td>14</td>
<td>11</td>
</tr>
</tbody>
</table>

Total number of records: 20000

The Field Scores area provides the following details:

- The Data Assessment service analyzed 20,000 records based on the Completeness and Conformance quality metrics.
- 19,745 records contain values for the Billing Street field.
- All 20,000 records contain values for the Created Date field.
- 14 records contain values for the Fax field.
- 11 records have values in the Fax field that conform.

A blank value in the Conformant column indicates that no conformance quality check was performed on the field. 0 appears in the Conformant column if no fields conform.

**Individual Scores**

The Individual Scores area of the scorecard provides an individual score for each quality metric included in the data assessment task.

The dashboard provides individual scores for the following quality metrics:

<table>
<thead>
<tr>
<th>Quality Metric</th>
<th>Description of Individual Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Validation</td>
<td>Percentage of address fields included in the quality metric that have valid addresses.</td>
</tr>
<tr>
<td>Completeness</td>
<td>Percentage of fields included in the quality metric that have values that are not null or blank.</td>
</tr>
<tr>
<td>Conformance</td>
<td>Percentage of fields included in the quality metric that conform.</td>
</tr>
<tr>
<td>Non-Duplicates</td>
<td>Percentage of records that are not duplicates.</td>
</tr>
</tbody>
</table>

**Overall Score**

The Overall Score area of the scorecard provides an overall score for the data assessment task. The overall score is the average of all individual scores for the data assessment task.

**Trend Chart**

The Trend Chart area of the scorecard shows the history of individual scores for each quality metric included in the data assessment task. Place the cursor over the lines in the chart to view individual scores for particular data quality jobs. If you run the data assessment task three times, you see individual scores for the quality metrics for each of the three jobs. You must run the data assessment task at least two times to see results in the trend chart.
Use the trend chart to monitor improvements or degradations in data quality over time. For example, you run a data assessment task and notice a large percentage of duplicate rows. You remove some of the duplicate rows, and then run the data assessment task again to determine the new percentage of duplicate rows. Use the trend chart to visually display the change in the duplicate scores.

Rules and Guidelines for Data Assessment Sources

Use the following rule and guideline for data assessment sources:

- Field names cannot contain spaces or hyphens.

Configuring a Data Assessment Task

Complete the following steps to configure a data assessment task:

1. Configure the source.
2. Configure quality metrics.
3. Optionally, configure data filters.
4. Optionally, configure scorecard thresholds.
5. Optionally, configure a schedule.

Step 1. Configure the Source

Configure a data assessment task for one Salesforce object. After you define the plan attributes, specify the Salesforce account, and then select the corresponding Salesforce object.

To configure the source:

1. To create a source, click Data Services > Data Assessment, and then click New.
2. To edit a source, click Data Services > Data Assessment, click the name of the data assessment task, and then click Edit.

2. On the Source page, update the following details:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Task Name      | Name of the data assessment task. The task names must be unique within the organization. Task names can contain alphanumeric characters, spaces, and the following special characters: _, *, -.
|                | Task names are not case sensitive.                               |
| Description    | Description of the data assessment task.                         |
3. Click Next.

### Step 2. Configure Quality Metrics

Add one or more of the following quality metrics to a data assessment task to analyze the quality of the Salesforce data:

- Address Validation
- Completeness
- Conformance
- Duplicates

When you include a quality metric in a plan, Informatica Cloud shows all fields included in the plan. All field names appear in Salesforce. When you add a Completeness, Conformance, or Duplicates quality metric, you can add or remove fields in the quality metric. Add a field to the quality metric to perform the quality check on the field. Fields that are greyed out cannot be added to a quality metric.

You may not be able to include a quality metric in a data assessment task for a particular Salesforce object. Some quality metrics are not applicable for all Salesforce objects. For example, you cannot include the Address Validation quality metric in a data assessment task for the Opportunity object. The Opportunity object does not contain address fields.

You must add at least one data quality metric to a data assessment task. Each data quality metric must have at least one field enabled for the quality check.

To add or change quality metrics:

1. To add a quality metric, click **Data Services > Data Assessment > New**, and then click **Quality Metrics**.
   
   To change a quality metric, click **Data Services > Data Assessment**, click the name of the task, click **Edit**, and then click **Quality Metrics**.

2. On the Quality Metrics page, update the following details:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicates</td>
<td>Adds the Duplicates quality metric to the plan.</td>
</tr>
<tr>
<td>Customize Weights</td>
<td>Customizes the weights for each field included in the Duplicates quality metric definition in the plan.</td>
</tr>
<tr>
<td>Completeness</td>
<td>Adds the Completeness quality metric to the plan.</td>
</tr>
<tr>
<td>Conformance</td>
<td>Adds the Conformance quality metric to the plan.</td>
</tr>
</tbody>
</table>
### Field Validation

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Validation</td>
<td>Adds the Address Validation quality metric to the plan.</td>
</tr>
<tr>
<td>Customize Field Mapping</td>
<td>Customizes the mapping between fields in Salesforce objects and fields in the data assessment task.</td>
</tr>
</tbody>
</table>

3. Click **Next**.

**Removing a Quality Metric**

To remove a quality metric:

- Clear the checkbox for the quality metric that you want to remove, and then click **Save**.

### Step 3. Configure Data Filters

You can add, change, or delete data filters from a data assessment task. Add data filters to filter the number of records on which you run the data assessment task. Delete a data filter to remove a restriction on the number of records processed by the data assessment task.

**Rules and Guidelines for Data Filters in Data Assessment Tasks**

Use the following rules and guidelines when you configure a data filter for a data assessment task:

- When you enter a number, you can include decimals, but not commas.
- When you write Salesforce data to a database target, verify that the Salesforce data uses the required formats for the following date and time datatypes:

<table>
<thead>
<tr>
<th>Datatype</th>
<th>Required Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>yyyy-MM-dd</td>
</tr>
<tr>
<td>DateTime</td>
<td>yyyy-MM-dd HH:mm:ss</td>
</tr>
</tbody>
</table>

If the Salesforce source contains a date and no time for the datetime datatype, the Data Loader service appends ‘00:00:00’ at the end of the date value to ensure the value is in the required format. When you write to a database target, the Data Loader service converts the Salesforce date and datetime data to the correct format expected by the database.

- The data filter must contain valid SQL or SOQL operators.
- When you include a Salesforce ID field in a filter, enter the exact ID value. If you enter a dummy ID value, the SOQL query fails.

**Configuring a Data Filter**

1. To add a data filter, click **Data Services > Data Assessment** and then click **Data Filters**.
2. To edit a data filter, click **Data Services > Data Assessment**, click the name of the task, click **Edit**, and then click **Data Filters**.
2. On the Data Filters page, enter the following details, and then click **Save**:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Limit</td>
<td>Select one of the following options:</td>
</tr>
<tr>
<td></td>
<td>- No Limit. Processes all records for the Salesforce object.</td>
</tr>
<tr>
<td></td>
<td>- Process Only the First. Processes the first X rows, where X is the number of rows. You might choose to process the first set of rows to test the task.</td>
</tr>
<tr>
<td>Drilldown</td>
<td>Select one of the following options:</td>
</tr>
<tr>
<td></td>
<td>- Enabled. Enables users to analyze Salesforce records with bad data quality based on the quality metrics included in the data assessment task.</td>
</tr>
<tr>
<td></td>
<td>- Disabled. Disables drilldown capability.</td>
</tr>
<tr>
<td>Data Filters</td>
<td>Click New to create a data filter. You can create simple or advanced data filters.</td>
</tr>
</tbody>
</table>

To delete a data filter, click the **Delete** icon next to the data filter.

3. Click **Next**.

**RELATED TOPICS:**
- “Data Filters” on page 145

### Deleting a Data Filter

To delete a data filter:

- Click the **Delete** icon next to the data filter that you want to delete, and then click **Save**.

#### Step 4. Configure Scorecard Thresholds

Configure scorecard thresholds to determine unacceptable, acceptable, and good data. The Data Assessment service assigns a score to a data assessment task and then rates the score based on the thresholds. The charts in the dashboard and Data Assessment Results page show the thresholds for the data assessment tasks.

#### Step 5. Configure a Schedule

You can run the data assessment task manually or schedule it to run at a specific time or on a time interval.

1. To add a schedule, click **Data Services > Data Assessment > New**, and then click **Schedule**.
   - To change a schedule, click **Data Services > Data Assessment**, click the name of the task, click **Edit**, and then click **Schedule**.
2. On the Schedule page, click **Run this task on schedule** and select the schedule you want to use.
   - To create a new schedule, click **New**. Enter schedule details and click **OK**.
   - To remove the task from a schedule, click **Do not run this task on a schedule**.
3. If you want to the scorecard for the task to display on the dashboard, click **True**. If you do not want the scorecard to display, click **False**.
4. Configure email notification options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Default Email Notification Options for my Organization</td>
<td>Use the email notification options configured for the organization.</td>
</tr>
</tbody>
</table>
| Use Custom Email Notification Options for this Task | Use the email notification options configured for the task. You can send email to different addresses based on the status of the task:  
- Error Email Notification. Sends email to the listed addresses when the task fails to complete.  
- Warning Email Notification. Sends email to the listed addresses when the task completes with errors.  
- Success Email Notification. Sends email to the listed addresses when the task completes without errors. Use commas to separate a list of email addresses. When you select this option, email notification options configured for the organization are not used. |

5. Click **Save**.

---

Customizing Weights

The Duplicates quality metric measures the percentage of duplicate records for a Salesforce object. The metric uses default weights and a default threshold to determine duplicate records.

Customize the weights assigned to fields to emphasize how significant the fields are in determining duplicate records. You can also change the threshold. By default, the threshold is 85.

Rules and Guidelines for Weights

Use the following rules and guidelines when you customize weights:

- The sum of all weights must equal 100.
- The Threshold Value must be an integer between 0 and 99, inclusive.
- Assign higher weights to fields that determine the record uniqueness.
  
  For example, the Salesforce Account object has a Billing Phone field and Billing State. Assign more weight to the Billing Phone field because it is more likely to uniquely identify an account record than the Billing State field.

- Assign higher weights to fields that have values.
  
  For example, a Salesforce Account object has Account Number and FAX fields. Each Salesforce Account record requires a unique account number, but does not require a fax number. Assign more weight to the Account Number field than to the FAX field. Not all records are guaranteed to have a FAX number.

- Assign no weight to fields that you want to exclude from the Duplicates quality metric.

- Specify a higher threshold to increase the requirements that determine whether records are duplicates.
  
  For example, you increase the matching accuracy if you increase the threshold from 60 to 80. However, the higher the threshold, the more likely the matching algorithm will consider similar values to be different. For example, if a street name is misspelled in one of two duplicate records and you configure a very high threshold, the matching algorithm may consider the records to be unique.
Customizing Field Mappings

By default, Informatica Cloud maps many standard and required fields of a Salesforce object to fields defined in the plan. You can customize the maps. Each data assessment task contains custom fields that you can use to map additional Salesforce source fields.

You can also reconfigure existing maps between fields. For example, you might store two phone numbers for each account in Salesforce. You store the main phone number for each account in the Phone field and the account contact phone number in a custom Phone2 field.

The following table shows the default mappings between Salesforce source fields and the plan fields:

<table>
<thead>
<tr>
<th>Salesforce Source Field</th>
<th>Data Assessment Task Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone</td>
<td>Phone</td>
</tr>
<tr>
<td>Phone2</td>
<td>&lt;not mapped&gt;</td>
</tr>
</tbody>
</table>

To perform a quality check on the phone number in the Phone2 field, you reconfigure the maps so that the Phone2 source field maps to the Phone field in the plan. You reconfigure the maps as shown in the following table:

<table>
<thead>
<tr>
<th>Salesforce Source Field</th>
<th>Data Assessment Task Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone</td>
<td>&lt;not mapped&gt;</td>
</tr>
<tr>
<td>Phone2</td>
<td>Phone</td>
</tr>
</tbody>
</table>

The Data Assessment service requires fields to be included in the field mapping because it sorts and groups records for each Salesforce object based on these fields. By default, the Data Assessment service maps the required fields. If you configure the field mappings, ensure that the field remains mapped.

The following table shows the required field for each applicable Salesforce object:

<table>
<thead>
<tr>
<th>Salesforce Object</th>
<th>Required Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account</td>
<td>BillingPostalCode</td>
</tr>
<tr>
<td>Contact</td>
<td>MailingPostalCode</td>
</tr>
<tr>
<td>Lead</td>
<td>PostalCode</td>
</tr>
<tr>
<td>Opportunity</td>
<td>Fiscal</td>
</tr>
</tbody>
</table>

Rules and Guidelines for Field Mappings

Use the following rules and guidelines when you customize field mappings:

- You can map only one Salesforce source field to any plan field.
- The Data Assessment service does not process plan fields that are not mapped.
Viewing Data Assessment Task Details

You can view details about a data assessment task, including the source connection name, Salesforce object on which it runs, quality metrics, data filters, scorecard thresholds, and schedule. When you view the details, click Edit to modify the plan.

- Click Data Services > Data Assessment, and then click the data assessment task name.

The details appear on the Data Assessment Tasks page.

**Note:** You can also monitor run-time details on the Home tab for a data assessment task that is running or that already completed.

Copying a Data Assessment Task

You can copy an existing data assessment task. You might copy a task to create a different task with similar behavior. For example, you might create two data assessment tasks to run the same quality metrics on two different Salesforce tables. Or, you might test changes on a copy of a task.

When you copy a task, the Data Assessment service creates a new task name. The Data Assessment service appends a number to the end of the task name. For example, if the original task name is DA1, the new task name is DA1_2.

**Note:** When you copy a task with schedule, the schedule is removed from the copied task.

To copy a data assessment task:

1. Click Data Services > Data Assessment.
2. Click the Make a Copy icon next to the task name.

**RELATED TOPICS:**
- “Rules and Guidelines for Copying Objects” on page 156

Running a Data Assessment Task

You can run a data assessment task in the following ways:

- Manually
- On a schedule

Running a Data Assessment Task Manually

You can start a data assessment task manually to run it immediately. You might run a data assessment task manually to test the task or to view the scorecard for the task immediately.

To run a data assessment task manually:

1. Click Data Services > Data Assessment.
2. On the Data Assessment Tasks page, click the Run icon for the data assessment task that you want to run.
Running a Data Assessment Task on a Schedule

Use a schedule to run the data assessment task at a specified time or at regular intervals. You can associate a schedule with one or more data assessment tasks. Edit a data assessment task to change the schedule associated with the plan.

1. To associate a schedule with a new task, click Data Services > Data Assessment, click New, and then click Schedule.
   To associate a schedule with an existing task, click Data Services > Data Assessment, select the data assessment task, click Edit, and then click Schedule.
2. Select a schedule in the Schedule field, or click New to create a schedule.

Stopping a Data Assessment Task

Click the Stop Job option in the activity monitor to stop a data assessment task that is currently running.

Monitoring a Data Assessment Task

You can view details about tasks that are currently running in the activity monitor. You can view details about completed tasks in the activity log.

Viewing the Scorecard for a Data Assessment Task

After you run a data assessment task, the Data Assessment service writes the results to a scorecard. You can view the scorecard on the following pages:

- Dashboard page
  Shows the scorecards for every data assessment task that ran and is configured to show results on the dashboard. It shows overall score, individual scores, and top issues. It shows up to five top issues.

- Data Assessment Results page
  Shows the scorecard for the specified data assessment task. The Data Assessment Results page shows more details than the dashboard. The Data Assessment Results page shows overall score, individual scores, trend chart, top issues, and field scores. It shows up to 20 top issues.

Analyzing Data Assessment Results

Scorecards show the results of data assessment tasks. Drill into the results to determine which Salesforce records cause bad quality scores in the scorecard. For example, analyze the scorecard results of an Address Validation data assessment task to determine which records have invalid address information.
When you analyze duplicate record results, the Data Assessment service generates and assigns duplicate IDs. Each duplicate ID uniquely identifies a set of duplicate records. The Data Assessment service shows the duplicate ID for each record in the results. Duplicate records have the same duplicate ID and are highlighted in the same color.

You can also download the results to a data file. The data file shows the duplicate ID, but does not use color to highlight records.

To analyze the data quality results:

1. Click **Home > Dashboard**.
   - Click **Data Services > Data Assessment**, and the click the **View Results** button for the data assessment task.
2. On the Dashboard or Data Assessment Results page, click one of the **Top Issues** links.
3. On the the Data Assessment Analysis page, select the fields of the records that you want to analyze.
4. Click **Fetch Data** to retrieve all records included in the data assessment task.
   - The results display.
5. To download the results to a data file, click the **Download to Excel** button or **Download Data to Excel** icon.
   - The Data Assessment service posts the data file on the Data Files page and notifies the creator of the plan that the download completed.
6. Click **Home > Data Files** to access the data file.
7. On the Data Files page, click the **Download File** icon to view or save the data file.
   - The Data Assessment service uses the following naming convention for the data files:

<table>
<thead>
<tr>
<th>Type of Quality Issue</th>
<th>Naming Convention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Validation</td>
<td>&lt;Data assessment task name&gt;<em>AV</em>&lt;Quality issue&gt;.csv</td>
</tr>
<tr>
<td></td>
<td>AddressValidation_Accounts_MC_AV_BillingAddress_Status.csv</td>
</tr>
<tr>
<td></td>
<td>AddressValidation_Accounts_MC_AV_ShippingAddress_Status.csv</td>
</tr>
<tr>
<td>Completeness</td>
<td>&lt;Data assessment task name&gt;<em>Completeness</em>&lt;Quality issue&gt;.csv</td>
</tr>
<tr>
<td></td>
<td>Completeness_Account_MC_Completeness_ShippingState.csv</td>
</tr>
<tr>
<td></td>
<td>Completeness_Account_MC_AV_ShippingAddress_Status.csv</td>
</tr>
<tr>
<td>Conformance</td>
<td>&lt;Data assessment task name&gt;<em>Conformance</em>&lt;Quality issue&gt;.csv</td>
</tr>
<tr>
<td></td>
<td>Conformance_Account_MC_Conformance_BillingCountry.csv</td>
</tr>
<tr>
<td>Duplicates</td>
<td>&lt;Data assessment task name&gt;_Duplicates.csv</td>
</tr>
<tr>
<td></td>
<td>DuplicateAccountRecords_Duplicates.csv</td>
</tr>
</tbody>
</table>
8. To delete a data file, click the **Delete** icon next to the data file.

### Deleting a Data Assessment Task

You can delete a data assessment task at any time. You cannot delete the task if it is included in a task flow. You must remove it from the task flow before you can delete the task.
Before you delete a data assessment task, verify that no users in the organization use it. You cannot retrieve a data assessment task after you delete it. When you delete a data assessment task, Informatica Cloud also deletes the corresponding scorecard from the dashboard and Data Assessment Results page.

To delete a data assessment task:

1. Click **Data Services > Data Assessment**.
2. Click the **Delete** icon next to the data assessment task name.
Data Loader Service

This chapter includes the following topics:

- Data Loader Service Overview, 82
- Rules and Guidelines for Data Loader Sources and Targets, 82
- Data Loader Prerequisite Tasks, 84
- Configuring a Data Loader Task, 84
- Viewing Data Loader Task Details, 89
- Running a Data Loader Task, 89
- Stopping a Data Loader Task, 90
- Monitoring a Data Loader Task, 90
- Deleting a Data Loader Task, 90

Data Loader Service Overview

The Data Loader service allows you to synchronize data between a source and target. For example, you can read updated contacts from your sales database and write them into Salesforce. You can also use expressions to transform the data according to your business logic, or use data filters to filter data before writing it to targets.

Complete the prerequisite tasks before you create a data loader task. When you create a task, you can associate it with a schedule to run it at specified times or on regular intervals. Or, you can run it manually. You can monitor tasks that are currently running in the activity monitor and view logs about completed tasks in the activity log.

Rules and Guidelines for Data Loader Sources and Targets

Use the following rules and guidelines for sources and targets in a data loader task:

- Each task must include a Salesforce source or target.
- Field names must contain 65 characters or less.
- Field names must contain only alphanumeric or underscore characters. Spaces are not allowed.
- Field names cannot start with a number.
- Each field name must be unique within each source and target object.
The service truncates the data if the scale or precision of a numeric target column is less than the scale or precision of the corresponding source column.

Rules and Guidelines for Salesforce Sources and Targets

Use the following rules and guidelines for Salesforce sources and targets in a data loader task:

- When you create a connection to a Salesforce source or target, the Data Loader service caches the connection metadata. If the connection metadata changes while you are creating the connection, you must log out and log in again to refresh the connection metadata.
- Use a Salesforce source or target in each task.
- The source must provide non-null values for required fields in a Salesforce target object.
- A data loader task may lose the least significant portion of numerical data in a Salesforce target field when the data uses most of the maximum precision and scale specified for the datatype of the field. For example, when trying to insert 65656565656565.6969 into a Salesforce field with datatype Decimal(14, 4), the task inserts 65,656,565,656,565.6950 instead. Or, when trying to insert 123456789123456789 into a Salesforce field with datatype Decimal(18, 0), the task inserts 123,456,789,123,456,784.
- A data loader task may round data unexpectedly into a Salesforce target field when the data uses most of the maximum precision and scale specified for the datatype of the field. For example, when trying to insert 555555555565.8855 into a Salesforce field with datatype Decimal(18, 0), the task inserts 555555555566 instead. However, when you manually enter the data in Salesforce, Salesforce rounds the data as 555555555565. Or, when trying to insert 12345678923456.3399 into a Salesforce field with datatype Decimal(14, 4), the task inserts 12,345,678,923,456.3400 instead.

Rules and Guidelines for Flat File Sources and Targets

Use the following rules and guidelines for flat file sources and targets in a data loader task:

- All date columns in a flat file source must have the same date format. The data loader task rejects rows that have dates in a different format than the one specified in the data loader task definition.
- Each flat file target must contain all fields that will be populated by the data loader task.
- The flat file cannot contain empty column names. If a file contains an empty column name, the following error appears:

  Invalid header line: Empty column name found.

- Column names in a flat file must contain printable ASCII (ASCII code 32 - 126) or tab characters. If the file contains an invalid character, the following error appears:

  Invalid header line: Non-printable character found. The file might be binary or might have invalid characters in the header line.

- If the column name of a flat file contains non-alphanumeric characters, starts with a number, or contains more than 75 characters, the Data Loader service modifies the column name. The Data Loader service truncates column names to 75 characters. For a flat file source, the data preview and expression dialog box shows the modified column names. For a flat file target, the Data Loader service changes the column name in the flat file when it generates the file during runtime.

Rules and Guidelines for Database Sources and Targets

Use the following rules and guidelines for relational sources and targets in a data loader task:

- You can use database tables, aliases, and views as sources. You can use database tables as a targets. You can also use an Oracle synonym as a source.
- Database sources and targets must meet the minimum system requirements.
A data loader task fails if it writes to an Oracle target field of a blob, bfile, or raw datatype. You cannot write source data to Oracle target fields with binary datatyps.

While reading time data from a MySQL source, the Data Loader service converts the time to the Coordinated Universal Time (UTC) time zone. Consider the time zone change if you create data filters and field mapping expressions based on time.

The database user account for each database target connection must have the DELETE, INSERT, SELECT, and UPDATE privileges.

**Note:** Instead creating or modifying relational tables, you can create aliases of the database tables to follow these guidelines.

### Data Loader Prerequisite Tasks

Before you create a data loader task, complete the following prerequisite tasks:

1. Create the flat file.
   You must create the flat file and its directory before you can create a data loader task that reads from or writes to a flat file.

2. Create the target database and table.
   To write Salesforce data to a database target, the database administrator must create a target database and table to store the Salesforce object data. The database must meet the minimum system requirements.

3. Create database users.
   To write Salesforce data to a database target, the database administrator must create a database user account in the target database. Each database user account must have the DELETE, INSERT, SELECT, and UPDATE privileges.

4. Verify that the sources and targets meet the requirements.

5. Download and run the Secure Agent.
   If you read data from or write data to a flat file or database, download and run a Secure Agent on a machine that runs the data loader task.

### Configuring a Data Loader Task

You can configure a data loader task using the Data Loader Task Wizard.

**Note:** You can also import a data loader task from Salesforce.

1. Define the data loader task.
2. Configure the source for the data loader task.
3. Configure the target for the data loader task.
4. Configure the data filters for the data loader task.
5. Map the source fields to the target fields.
6. Configure the schedule for the data loader task.
Step 1. Define the Data Loader Task

Define a data loader task.

1. To define a data loader task, click Data Services > Data Loader, and then click New.
   To edit the definition of an existing data loader task, click Data Services > Data Loader, click the name of the data loader task, and then click Edit.

2. In the Data Loader Task Details area, configure the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Name</td>
<td>Name of the data loader task. The names of data loader tasks must be unique within the organization. Data loader task names can contain alphanumeric characters, spaces, and the following special characters: _ . + - . Data loader task names are not case sensitive.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the data loader task. Maximum length is 255 characters.</td>
</tr>
</tbody>
</table>
| Task Operation | Select one of the following task operation types:  
- Insert  
- Update  
- Upsert  
- Delete  
The list of available targets in a subsequent step depend on the task operation you select.                                      |

3. Click Next.

Step 2. Configure the Source

Select the source for the data loader task.

1. To configure a source for a new data loader task, click Data Services > Data Loader > New, complete the preliminary wizard steps, and then click Source.
   To configure a source for an existing data loader task, click Data Services > Data Loader, click the name of the data loader task, click Edit, and then click Source.

2. On the Source page, select a connection.
   Click New to create a connection or click Edit to edit a connection.

3. For Source Object, select a source.
   When you select a source, the Data Preview area shows the first ten rows of the first five columns in the source. It also displays the total number of columns in the source.
   If the source data contains binary data, the Data Preview area shows the following text:
   BINARY DATA

4. To preview all source columns in a flat file, click Preview All Columns.
   The flat file shows the first ten rows of the source.

5. Click Next.

Step 3. Configure the Target

Select the target for the data loader task.
Available options depend on the source type and task operation you select for the task. For example, if you select a flat file source connection type, you must select a Salesforce target connection type. If you select the upsert task operation, you cannot select a flat file target connection because you cannot upsert records into a flat file target.

If the data loader task writes to a Salesforce object target using the update or upsert task operation and the Salesforce objects use external IDs, specify the external ID for each related object.

To configure a target:

1. To configure a target for a new data loader task, click Data Services > Data Loader > New, complete the preliminary wizard steps, and then click Target.
   To configure a target for an existing data loader task, click Data Services > Data Loader, click the name of the data loader task, click Edit, and then click Target.

2. On the Target page, enter the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Select a connection. The list of available connections depends on source type and the task operation defined for the task. Click New to create a connection or click Edit to edit a connection.</td>
</tr>
<tr>
<td>Target Object</td>
<td>Select the target object to which you want to write data. You must create the flat file before you can select it.</td>
</tr>
<tr>
<td>Truncate Target</td>
<td>Database targets with the Insert task operation only. Truncates a database target table before inserting new rows. Select True to truncate the target table before inserting all rows. Select False to insert new rows into existing target table without truncating the table. Default is False. When Informatica Cloud truncates a target through an ODBC connection, it executes a DELETE FROM statement to truncate the table. When Informatica Cloud truncates a target using other connection types, it executes a TRUNCATE TABLE statement.</td>
</tr>
</tbody>
</table>

When you select a target, the Data Preview area shows the first ten rows of the first five columns in the target. It also displays the total number of columns in the target.

If the source data contains binary data, the Data Preview area shows the following text:

```
BINARY DATA
```

3. To preview all target columns in a flat file, click Preview All Columns.
   The flat file shows the first ten rows of the target.

4. Click Next.

### Step 4. Configure Data Filters

Use a data filter to reduce the number of source rows that the Data Synchronization service reads for the task. By default, the Data Synchronization service reads all source rows.

1. To add a data filter to a new data loader task, click Data Services > Data Loader > New, complete the preliminary wizard steps, and then click Data Filters.
   To add or change a data filter in an existing data loader task, click Data Services > Data Loader, click the name of the data loader task, click Edit, and then click Data Filters.
On the Data Filters page, enter the following details, and then click **Save**:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Limit</td>
<td>Select one of the following options:</td>
</tr>
<tr>
<td></td>
<td>- No Limit. Processes all records for the Salesforce object.</td>
</tr>
<tr>
<td></td>
<td>- Process Only the First. Processes the first X rows, where X is the number of rows. You might choose to process the first set of rows to test the task. You cannot specify a row limit on data loader tasks with non-Salesforce sources. If you select a non-Salesforce source, the option is greyed out.</td>
</tr>
<tr>
<td>Data Filters</td>
<td>Click New to create a data filter on a Salesforce or database source. You can create simple or advanced data filters.</td>
</tr>
</tbody>
</table>

3. Click the **Delete** icon next to the data filter to delete the filter.

4. Click **Next**.

**Related Topics:**
- “Data Filters” on page 145
- “Rules and Guidelines for Data Filters” on page 147

**Step 5. Configure the Field Mapping**

Map source columns to target columns on the Field Mapping page of the Data Loader Task wizard. You must map at least one source column to a target column. Map columns with compatible datatypes or create field expressions to convert datatypes appropriately.

Depending on the task operation, the Data Loader service requires certain fields to be included in the field mapping. By default, the Data loader service maps the required fields. If you configure the field mapping, ensure that the required fields remain mapped. If you do not map the required fields, the data loader task fails.

The following table shows the required fields for each applicable task operation and target type:

**Table 3. Required Fields in Field Maps for Task Operations and Target Types**

<table>
<thead>
<tr>
<th>Required Field</th>
<th>Task Operations</th>
<th>Target Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Delete</td>
<td>Salesforce</td>
<td>Map the ID column to enable the Data Loader service to identify records to delete or update in a Salesforce target.</td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Upsert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>External ID</td>
<td>Update</td>
<td>Salesforce</td>
<td>Map at least one External ID field to enable the Data Loader service to identify records to upsert in a Salesforce target.</td>
</tr>
<tr>
<td>Primary Keys</td>
<td>Delete</td>
<td>Database</td>
<td>Map primary key columns to enable the Data Loader service to identify records to delete, update, or upsert in a database target.</td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Upsert</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-null fields</td>
<td>Insert</td>
<td>Salesforce</td>
<td>Map all fields that cannot be null in Salesforce.</td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Upsert</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To configure a field mapping:

1. To configure a field mapping in a new data loader task, click **Data Services > Data Loader > New**, complete the preliminary wizard steps, and then click **Field Mapping**.
   - To configure a field mapping in an existing data loader task, click **Data Services > Data Loader**, click the name of the data loader task, click **Edit**, and then click **Field Mapping**.

2. On the Field Mapping page, select and drag the source fields to the applicable target fields.
   - If the fields do not appear or do not appear correctly for a source or target, click **Refresh Fields**. If you change the source or target, Informatica Cloud does not automatically refresh the fields. Informatica Cloud caches the field metadata. Refresh the fields to update the cache and view the latest field attributes of a source or target.
     - **Note**: Fields can be refreshed when the database server for the database source or target and the Secure Agent are all running.

3. Click the following options to perform additional tasks:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-Match</td>
<td>Click to match source fields to similarly named target fields. By default, the Data Loader Task Wizard maps similarly named fields. If you make changes to the default, click this option to remap similarly named fields.</td>
</tr>
<tr>
<td>Clear Mapping</td>
<td>Click to clear all field maps.</td>
</tr>
<tr>
<td>Add Expression</td>
<td>Click to transform the source data before loading it into the target field. For more information, see “Field Expressions” on page 141.</td>
</tr>
<tr>
<td>Validate Mapping</td>
<td>Validate the field mapping. For more information, see “Rules and Guidelines for Validating Field Mappings” on page 141.</td>
</tr>
<tr>
<td>Related Objects</td>
<td>Salesforce only. Click to specify the external ID of each related object if you use external IDs to uniquely identify records in related Salesforce objects. In the Related Objects page, select the external ID for each applicable related object. You do not need to specify the external ID for a related object if you do not want to update changes for that related object. If you edit an existing task and select a new target, before you can select related objects, click Refresh Fields in step 5 to refresh the fields. If you do not specify the external ID, the Data Loader service requires that the source provide the Salesforce ID to uniquely identify records in each related object.</td>
</tr>
</tbody>
</table>

4. To delete a field mapping, select the **Clear expression/lookup** icon for the target field.

5. **Click Next**.

**RELATED TOPICS:**
- “IDs for Related Salesforce Objects” on page 139
- “Rules and Guidelines for Datatypes” on page 177

**Step 6. Configure a Schedule**

You can run a data loader task manually or schedule it to run at a specific time or on a time interval.

1. To add a schedule in a new data loader task, click **Data Services > Data Loader > New**, complete the preliminary wizard steps, and then click **Schedule**.
   - To add or change a schedule in an existing data loader task, click **Data Services > Data Loader**, click the name of the data loader task, click **Edit**, and then click **Schedule**.
2. On the Schedule page, click **Run this task on schedule** and select the schedule you want to use.
   - To create a new schedule, click **New**. Enter schedule details and click **OK**.
   - To remove the task from a schedule, click **Do not run this task on a schedule**.
3. Click **Save**.

### Viewing Data Loader Task Details

You can view details about a data loader task, including the source and target connections and the associated schedule. When you view the details, click **Edit** to modify the data loader task. You can also view the activity log for a data loader task.

- Click **Data Services > Data Loader**, and then click the data loader task name.

### Running a Data Loader Task

You can run a data loader task in the following ways:

- Manually
- On a schedule

### Rules and Guidelines for Running a Data Loader Task

Use the following rules and guidelines when you run a data loader task:

- To run a data loader task that reads data from or writes data to a database, the Secure Agent and the corresponding database server must be running. The data loader task fails if either is not running.
- Ensure that the source and target definitions are current. If the source or target no longer contains fields that are mapped in the field mapping, the data loader task fails.
- You cannot run multiple instances of a data loader task simultaneously. If you run a data loader task that is already running, the data loader task fails with the following error:
  
  Failed to process any data rows. Data loader task `<data loader task name>` failed to run. Another instance of the task is currently executing.
  
  If you configured the data loader task to run on a schedule, increase the time interval between the scheduled tasks to prevent multiple instances of the data loader task from running simultaneously. If you run the data loader task manually, wait for the currently running instance of the data loader task to complete before starting it again. You can view currently running data loader tasks in the activity monitor.

### Running a Data Loader Task Manually

Run a data loader task manually to immediately start the data loader task. You might want to run a data loader task manually for the following reasons:

- To verify that the data loader task is configured properly.
- To load data into the target occasionally. You may not need to update the target on regular intervals.

When you start a data loader task manually, the activity monitor appears and shows you details about the task.
To run a data loader task manually:
1. Click Data Services > Data Loader.
2. On the Data Loader Tasks page, click the Run icon for the data loader task that you want to run.

Running a Data Loader Task on a Schedule

Use a schedule to write data at a specified time or on regular intervals. Edit a data loader task to change the schedule associated with it.

1. To associate a schedule with a new data loader task, click Data Services > Data Loader > New, complete the preliminary wizard steps, and then click Schedule.
   To associate a schedule with an existing data loader task, click Data Services > Data Loader, click the name of the data loader task, click Edit, and then click Schedule.
2. Select a schedule in the Schedule area, or click New to create a schedule.

Stopping a Data Loader Task

Click the Stop Job option in the activity monitor to stop a data loader task that is currently running.

Monitoring a Data Loader Task

You can view details about tasks that are currently running in the activity monitor. You can view details about completed tasks in the activity log.

Deleting a Data Loader Task

You can delete a data loader task at any time. If the data loader task is currently running, the Data Loader service completes the task and cancels all scheduled tasks that have not started.

Before you delete a data loader task, verify that no users in the organization plan to use it. You cannot retrieve a data loader task after you delete it.

To delete a data loader task:
1. Click Data Services > Data Loader.
2. Click the Delete icon next to the data loader task name.
CHAPTER 6

Data Replication Service

This chapter includes the following topics:

- Data Replication Service Overview, 91
- Data Replication Sources and Targets, 92
- Rules and Guidelines for Data Replication Sources and Targets, 92
- Data Replication Prerequisite Tasks, 93
- Configuring a Data Replication Task, 93
- Viewing Data Replication Task Details, 99
- Copying a Data Replication Task, 100
- Running a Data Replication Task, 100
- Stopping a Data Replication Task, 102
- Monitoring a Data Replication Task, 102
- Deleting a Data Replication Task, 102
- Understanding Table and Column Names in a Database Target, 102
- Resetting a Database Target, 104
- Creating Target Tables, 105
- Understanding Load Types, 105

Data Replication Service Overview

The Data Replication service allows you to replicate data to a target. You might replicate data to back up the data or perform offline reporting. You can replicate data in Salesforce objects or database tables to databases, or flat files.

You can perform the following tasks for data replication tasks:

- Create a data replication task.
- Edit a data replication task.
- View data replication task details.
- Run a data replication task.
- Stop a data replication task.
- Monitor a data replication task.
- Delete a data replication task.
Data Replication Sources and Targets

You can replicate Salesforce and database data to the following types of targets:

- Database
- Flat file

Replicating Salesforce Sources

When you replicate Salesforce sources, you can replicate all current rows in the Salesforce source. You can also replicate deleted and archived rows in Salesforce sources. You might replicate deleted or archived rows to preserve or analyze historical data.

To replicate all historical data for a Salesforce source, on the Sources page of the Data Replication wizard, select Include Archived and Deleted Rows in the Source.

The Data Replication service does not replicate Salesforce Vote or UserProfileFeed objects. Data replication tasks configured to replicate All Objects do not include Vote or UserProfileFeed objects. When you select Salesforce objects for replication, the Vote and UserProfileFeed objects are not available in the Objects to Replicate dialog box.

Replicating Data to a Database Target

When you run the data replication task for the first time, the Data Replication service creates the database tables, and then writes the replicated data to the tables. During subsequent runs, the Data Replication service behavior depends on the load type. If the data replication task is configured with full load, the Data Replication service truncates the database tables, and then writes the source data to the tables. If the data replication task is configured with incremental load, the Data Replication service overwrites existing records in the database tables with changed and new records.

If you run a data replication task that replicates Salesforce data to a database target, the Data Replication service replicates data for each Salesforce object in the data replication task to a separate database table.

Replicating Data to a Flat File Target

When you run the data replication task for the first time, the Data Replication service creates the flat files, stores them in the specified directory, and then writes the replicated data to the files. During subsequent runs, the Data Replication service updates the existing flat files with new or changed data.

If you run a data replication task that replicates data to a flat file target, the Data Replication service replicates data for each Salesforce object in the data replication task to a separate flat file.

Rules and Guidelines for Data Replication Sources and Targets

Use the following rules and guidelines for sources and targets in a data replication task:

- You cannot configure multiple data replication tasks to replicate the same source object to the same target object. For example, you cannot configure two data replication tasks to write Salesforce Account data to the SF_ACCOUNT Oracle database table.
You cannot simultaneously run multiple data replication tasks that write to the same target table.

- If the datatype, precision, or scale of a Salesforce field changes, you must run a full load to replicate Salesforce data to a target. If you run an incremental load, the task fails.
- The corresponding database server must be running for a data replication task to access a relational source or target.
- You can use database tables as targets. However, you can use database tables, aliases, and views as sources. You can also use an Oracle synonym as a source.
- The Data Replication service does not detect field length changes in the source after you run the data replication task for the first time. As a result, the task fails if the source field length is longer than the target field length.
- If you configure a data replication task to replicate all objects from the Public schema of an Oracle database and the schema contains synonyms, the task may fail. The task fails when the user account for the Oracle connection does not have the Select privilege on the synonyms in the Public schema or the synonym names do not begin with a letter or underscore (_).

### Data Replication Prerequisite Tasks

Complete the following prerequisite tasks before you create a data replication task:

1. **Create a database.**
   - To replicate data to a database target, the database administrator must create a target database. The database must meet the minimum system requirements.

2. **Create database users.**
   - To replicate data to a database target, the database administrator must create a database user account in the target database. Each database user account must have the CREATE, DELETE, DROP, INSERT, SELECT, and UPDATE privileges. You must have a database user account for each data replication task that writes to that database. You can use the same database user account for multiple data replication tasks. If you use the same database user account for multiple groups, ensure that the data replication tasks do not overwrite data in the same target tables.

3. **Create a directory for the flat files.**
   - To replicate data to a flat file, create a directory to store the flat files.

4. **Download and run the Secure Agent.**
   - To replicate data to a flat file or database target, you must download and run at least one Secure Agent on a machine in your network.

5. **Optionally, create source and target connections.**

6. **Optionally, create a schedule.**
   - Create a schedule to run data replication tasks at specified times or on regular intervals.

### Configuring a Data Replication Task

Create a data replication task to replicate data from a source to a target. When you create a data replication task, you specify the source connection, target connection, and objects to replicate. A data replication task can replicate...
data from one or more Salesforce objects or database tables. You can also exclude rows and columns from the
data replication task. Associate a schedule with a data replication task to specify when the Data Replication service replicates the data and how often it repeats the replication.

You configure a data replication task to run full or incremental loads. Perform a full load to replicate all rows of
data for each specified object. Perform an incremental load to replicate rows that are new or changed since the last time you ran the data replication task.

Edit a data replication task to update the data replication task details, source or target connection, objects to replicate, or rows or columns to replicate. You might also edit a data replication task to add the data replication task to a schedule or remove the data replication task from a schedule. If you remove the data replication task from a schedule and the data replication task is currently running, the Data Replication service completes the replication. The Data Replication service cancels the remaining scheduled data replication tasks that have not started.

To configure a data replication task using the Data Replication Task Wizard, complete the following steps:

1. Configure the source.
2. Configure the target.
3. Optionally, exclude fields.
4. Optionally, configure data filters.
5. Optionally, configure a schedule.

**Rules and Guidelines for Configuring Data Replication Tasks**

Use the following rules and guidelines for configuring data replication tasks:

- The names of source tables and fields can contain at most 79 characters.
- Multiple data replication tasks cannot write to the same database table or flat file.
- You cannot configure a data replication task with the same source and target objects. If the source and target connections are the same, you must enter a target prefix to distinguish the source and target objects.
- You cannot configure a data replication task to replicate the Vote or UserProfileFeed objects.
- You cannot replicate data to a Salesforce target.
- The maximum number of characters that a data replication task can write to each row in a Microsoft SQL Server 2000 target is 8060. If a data replication task tries to write more than the maximum amount of characters to a row, the task fails with the following error:

```text
WRT_8229 Database errors occurred: FnName: Execute -- [Microsoft][ODBC SQL Server Driver][SQL Server]Cannot create a row of size <row size> which is greater than the allowable maximum of 8060. FnName: Execute -- [Microsoft][ODBC SQL Server Driver][SQL Server]The statement has been terminated.
```

**Step 1. Configure the Source**

Column names of a database source must not contain spaces or hyphens.

1. To define the source for a new data replication task, click **Data Services > Data Replication**, and then click **New**.
   
   To define the source for an existing data replication task, click **Data Services > Data Replication**, click the name of the data replication task, and then click **Edit**.
2. On the Source page, in the Task Details area, configure the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Name</td>
<td>Name of the data replication task. Task names must be unique within the organization. Task names can contain alphanumeric characters, spaces, and the following special characters: _ + -. Task names are not case sensitive.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the task. Maximum length is 255 characters.</td>
</tr>
</tbody>
</table>

3. In the Source Details area, select a connection.
   - Click **New** to create a connection or click **Edit** to edit a connection.

4. To replicate all objects in the database or Salesforce account, select **All Objects**.
   - If you run an incremental load, the Data Replication service replicates data for new objects that did not exist during previous jobs.
   - Or, to select objects to replicate, click **Select**.
     - In the Objects to Replicate dialog box, select the objects you want to replicate and click **OK**.
     - **Note:** For Salesforce, only Salesforce objects that can be queried appear in the Objects to Replicate area. If an object does not appear, contact the Salesforce administrator.

5. If you want the Data Replication service to stop processing the task when it encounters an error, click **Cancel processing the remaining objects**.
   - If you want the Data Replication Service to continue to process a task after it encounters an error, click **Continue processing of the remaining objects**.
   - By default, the Data Replication service stops processing the task when it encounters an error.

6. For Salesforce sources, if you want the Data Replication service to replicate historical data in Salesforce sources, including archived and deleted rows, select **Include archived and deleted rows in the source**.
   - By default, the Data Replication service replicates only current source rows in selected Salesforce sources and ignores archived and deleted rows.

7. Click **Next**.

**Step 2. Configure the Target**

Configure a target for the data replication task.

**Target Prefixes**

When you replicate data to a database table or flat file, the Data Replication service names each database table or flat file based on the corresponding source object name. By default, the data replication task includes the target prefix SF_. For example, the default flat file name for the Account Salesforce object is SF_ACCOUNT.CSV. If you remove the default target prefix and do not specify another prefix, the Data Replication service creates a flat file or database table with the same name as the corresponding source object.

You can use target prefixes to prevent overwriting data. For example, you and another user share a database user account. The other user ran a data replication task on the Contact object from her Salesforce account. Her data replication task created a database table named Contact in the shared database. You elect to use no target prefix and run a data replication task on the Contact object from your Salesforce account. The Data Replication service overwrites the data in the existing Contact table with your data. If you use the SF_ prefix, the Data Replication service creates a table named SF_CONTACT and does not overwrite the existing table named Contact.
**Note:** If you configure a prefix for the target table name, ensure that the prefix and corresponding Salesforce object name do not exceed the maximum number of characters allowed for a target table name. For more information, see “Truncating Table Names” on page 103.

## Configuring a Target

1. To define the target for a new data replication task, click **Data Services > Data Replication > New**, complete the preliminary wizard steps, and then click **Target**.

   To define the target for an existing data replication task, click **Data Services > Data Replication**, click the name of the data replication task, click **Edit**, and then click **Target**.

2. On the Target page, enter the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Select a connection. Click New to create a connection or click Edit to edit a connection. Click Test to test the connection.</td>
</tr>
<tr>
<td>Target Prefix</td>
<td>Prefix that is added to Salesforce object names to create the flat file names or table names in a target database.</td>
</tr>
</tbody>
</table>
| Load Criteria  | Type of load. Select one of the following options:
|                | - Incremental Load. Loads changed data only.                                                                                                   |
|                | - Full Load. Loads all data.                                                                                                                     |
|                | This option is enabled for tasks with a Salesforce source and a relational target. For all other tasks, the Data Replication service performs a full load. |
| Delete Criteria| Select one of the following options:
|                | - Remove Deleted Columns and Rows. Deletes columns and rows from the target if they no longer exist in the source.                          |
|                | - Retain Deleted Columns and Rows. Retains columns and rows in the target that were removed from the source.                                   |

3. Click **Next**.

**RELATED TOPICS:**
- “Target Prefixes” on page 95
- “Understanding Load Types” on page 105

## Step 3. Configure the Field Exclusions

By default, the data replication task loads all fields into the target. Configure field exclusions for each source object to limit the fields loaded into a target.

1. To configure the field exclusions for a new data replication task, click **Data Services > Data Replication > New**. Complete the preliminary wizard steps, and then click **Field Exclusions**.

   To configure the field exclusions for an existing data replication task, click **Data Services > Data Replication**, click the name of the data replication task, click **Edit**, and then click **Field Exclusions**.

2. On the Field Exclusion page, click **Exclude Fields** to select fields to exclude.

3. Click **Next**.
Excluding Fields
You can specify whether to exclude fields from each object included in a data replication task.
1. Click Exclude Fields in the Field Exclusions page of the Data Replication Task Wizard.
2. In the Field Exclusion window, select the source object. The fields of the object appear.
3. To exclude a field, move the field to the Excluded Fields area.
4. To include a field, move the field to the Included Fields area.
5. Click OK.

Step 4. Configure the Data Filters
By default, the data replication task replicates all source rows to the target. Configure data filters to filter source rows that are replicated. If you replicate multiple source objects, create a different set of data filters for each object.
1. To configure the data filters for a new data replication task, click Data Services > Data Replication > New, complete the preliminary wizard steps, and then click Data Filters.
   To configure the data filters for an existing data replication task, click Data Services > Data Replication, click the name of the data replication task, click Edit, and then click Data Filters.
2. On the Data Filters page, enter the following details:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Limit</td>
<td>Select one of the following options:</td>
</tr>
<tr>
<td></td>
<td>- No Limit. Replicates all rows of the source.</td>
</tr>
<tr>
<td></td>
<td>- Process Only the First. Replicates the first X rows, where X is the number of rows. You might choose to process the first set of rows to test the task. You cannot specify a row limit on data replication tasks with non-Salesforce sources. If you select a non-Salesforce source, the option is greyed out.</td>
</tr>
<tr>
<td>Data Filters</td>
<td>Click New to create a data filter on a Salesforce or database source. You can create simple or advanced data filters.</td>
</tr>
</tbody>
</table>
3. Click the Delete icon next to the data filter to delete the filter.
4. Click Next.

Related Topics:
- “Data Filters” on page 145

Step 5. Configure a Schedule
You can run a data replication task manually or schedule it to run at a specific time or on a time interval.

Understanding Email Notification Options
You can configure email notification for a Data Synchronization or Data Replication task. When you configure email notification for the task, Informatica Cloud uses the email notification options configured for the task instead of the email notification options configured for the organization. You can send email to different addresses based on the status of the task:
- Success. The task completed successfully.
Warning. The task completed with errors.
Error. The task did not complete.

Understanding Preprocessing and Postprocessing Commands
You can run preprocessing and postprocessing commands to perform additional tasks. The service runs preprocessing commands before it reads the source. It runs postprocessing commands after it writes to the target.

You can use the following types of commands:
- SQL commands. Use SQL commands to perform database tasks.
- Operating system commands. Use shell and DOS commands to perform operating system tasks.

If any command in the preprocessing or postprocessing scripts fail, the service fails the task.

Rules and Guidelines for Preprocessing and Postprocessing SQL Commands
You can run SQL commands before or after a task. For example, you can use SQL commands to drop indexes on the target before the task runs, and then recreate them when the task completes.

Use the following rules and guidelines when creating the SQL commands:
- Use any command that is valid for the database type. However, Informatica Cloud does not allow nested comments, even if the database allows them.
- Use a semicolon (;) to separate multiple statements. Informatica Cloud issues a commit after each statement.
- Informatica Cloud ignores semicolons within /* ... */.
- If you need to use a semicolon outside of comments, you can escape it with a backslash (\).
- Informatica Cloud does not validate the SQL.

Rules and Guidelines for Preprocessing and Postprocessing Operating System Commands
Informatica Cloud can perform operating system commands before or after the task runs. For example, use a preprocessing shell command to archive a copy of the target flat file before the task runs on a UNIX machine.

You can use the following types of operating system commands:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Command Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX</td>
<td>Any valid UNIX command or shell script.</td>
</tr>
<tr>
<td>Windows</td>
<td>Any valid DOS or batch file.</td>
</tr>
</tbody>
</table>

Configuring a Schedule
To configure a schedule for a data replication task:

1. To configure a schedule for a new data replication task, click **Data Services > Data Replication > New**, complete the preliminary wizard steps, and then click **Schedule**.
   - To configure a schedule for an existing data replication task, click **Data Services > Data Replication**, click the name of the data replication task, click **Edit**, and then click **Schedule**.
2. On the Schedule page, click **Run this task on schedule** and select the schedule you want to use.
To create a new schedule, click **New**. Enter schedule details and click **OK**.

To remove the task from a schedule, click **Do not run this task on a schedule**.

3. Configure email notification options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Default Email Notification Options for my Organization</td>
<td>Use the email notification options configured for the organization.</td>
</tr>
</tbody>
</table>
| Use Custom Email Notification Options for this Task | Use the email notification options configured for the task. You can send email to different addresses based on the status of the task:  
- **Error Email Notification**. Sends email to the listed addresses when the task fails to complete.  
- **Warning Email Notification**. Sends email to the listed addresses when the task completes with errors.  
- **Success Email Notification**. Sends email to the listed addresses when the task completes without errors.  
Use commas to separate a list of email addresses.  
When you select this option, email notification options configured for the organization are not used. |

4. Optionally, enter the following advanced options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing Commands</td>
<td>Commands to run before the task.</td>
</tr>
<tr>
<td>Postprocessing Commands</td>
<td>Commands to run after the task completes.</td>
</tr>
</tbody>
</table>

5. Click **Save**.

---

**Viewing Data Replication Task Details**

You can view details about a data replication task, including the load criteria, source and target connections, field exclusions, data filters, and the associated schedule. When you view the details, click **Edit** to modify the data replication task.

- Click **Data Services > Data Replication**, and then click the data replication task name.

  The details appear on the View Data Replication Task page.
Copying a Data Replication Task

You can copy a data replication task. You might copy a task to create a different task with similar behavior. For example, you might create two data replication tasks to replicate the same data to two different targets. Or, you might test changes on a copy of a task.

When you copy a task, the Data Replication service creates a new task name and, if applicable, target prefix. The Data Replication service appends a number to the end of the task name and target prefix. For example, if the original task name is RG1 and target prefix is SF_, the new task name is RG1_2 and target prefix is SF2_.

**Note:** When you copy a task with a schedule, the schedule is removed from the copied task.

To copy a data replication task:

1. Click **Data Services > Data Replication**.
2. Click the **Make a Copy** icon next to the task name.

**Related Topics:**

- “Rules and Guidelines for Copying Objects” on page 156

Running a Data Replication Task

You can run a data replication task in the following ways:

- **Manually**
  
  Runs a data replication task at any time. When you run a data replication task manually, the data replication task starts immediately.

- **On a schedule**
  
  Configure a data replication task to run on a schedule to replicate data from Salesforce objects to the target database at a specified time. When you configure a data replication task to run on a schedule, include a repeat frequency to replicate the data at regular intervals.

Rules and Guidelines for Running a Data Replication Task

Use the following guidelines when you run a data replication task:

- The Secure Agent and the target database server must be running to run the data replication task. The data replication task fails if either is not running.

- You cannot run multiple instances of a data replication task simultaneously. If you run a data replication task that is already running, the data replication task fails with the following error:

  ```
  Data replication task failed to run. Another instance of task <data replication task name> is currently replicating the same objects.
  ```

If you configured the data replication task to run on a schedule, increase the time interval between the scheduled jobs to prevent multiple instances of the data replication task from running simultaneously. If you run the data replication task manually, wait for the currently running instance of the data replication task to complete before starting it again. You can view currently running data replication tasks in the activity monitor.
You cannot simultaneously run data replication tasks that replicate the same Salesforce object. If you simultaneously run two data replication tasks that replicate the same object, the data replication task that starts first obtains a lock on the Salesforce object, and the other data replication task fails with the following error:

```
Data replication task failed to run. Data replication task <data replication task name> is currently replicating the same objects.
```

If you configured the data replication tasks to run on schedules, schedule the data replication tasks to run at different times. If you run the data replication tasks manually, wait until one completes before you run the other.

- If the length of the source object name exceeds the maximum number of characters allowed for a table name in a relational target, the Data Replication service truncates the corresponding table name.
- If the length of the column name of the source object exceeds the maximum number of characters allowed for a column in a relational target, the Data Replication service truncates the corresponding column name.
- If you run a data replication task that excludes fields and that writes to a database target, the Data Replication service drops any indexes defined on the excluded fields in the target table.
- If you replicate timestamp data, the Data Replication service truncates the millisecond portion of the timestamp data.
- If you replicate timestamp data to an Oracle database, the Data Replication service truncates fractional seconds to the second. For example, the Data Replication service truncates 12:52:47.58 to 12:52:47.
- You can load data from Salesforce fields of any datatype, except binaries, into a flat file target.
- You cannot run a data replication task with an incremental load if the target is a flat file or the source is not Salesforce.

### Running a Data Replication Task Manually

Run a data replication task manually to immediately start the data replication task. You might want to run a data replication task manually for the following reasons:

- To verify that a data replication task is configured properly.
- To replicate the data occasionally. You may not want to replicate data at regular intervals.

When you start a data replication task manually, the activity monitor appears and shows you details about the replication.

To run a data replication task manually:

1. Click **Data Services > Data Replication**, and then select **List View**.
2. On the list view of the Data Replication Tasks page, click the **Run** icon for the data replication task that you want to run.

### Running a Data Replication Task on a Schedule

Use a schedule to replicate data at a specified time or at regular intervals. You can associate a schedule with one or more data replication tasks. Edit a data replication task to change the schedule associated with it.

1. To associate a schedule with a new data replication task, click **Data Services > Data Replication**, click **New**, complete the preliminary wizard steps, and then click **Schedule**.
   
   To associate a schedule with an existing data replication task, click **Data Services > Data Replication**, click the data replication task name, click **Edit**, and then click **Schedule**.
2. Select a schedule in the Schedule area, or click **New** to create a schedule.
Stopping a Data Replication Task

Click the Stop Job option in the activity monitor to stop a data replication task that is currently running.

Monitoring a Data Replication Task

You can view details about tasks that are currently running in the activity monitor. You can view details about completed tasks in the activity log.

Deleting a Data Replication Task

Delete a data replication task if you no longer need to replicate the source objects assigned to the data replication task. You can delete a data replication task at any time. If the data replication task is currently running, the Data Replication service completes the replication and cancels scheduled data replication tasks that have not started.

You cannot delete the task if it is included in a task flow. You must remove it from the task flow before you can delete the task. Before you delete a data replication task, verify that no users in the organization plan to use it. You cannot retrieve a data replication task after you delete it.

To delete a data replication task:

1. Click **Data Services** > **Data Replication**, and then select **List View**.
2. Click the **Delete** icon next to the data replication task name.

Understanding Table and Column Names in a Database Target

The Data Replication service replicates source objects and fields to target database tables and columns, respectively. In certain cases, the Data Replication service does not give the target table and column names the same names as the source objects and fields.

The Data Replication service may not use the same names in the following circumstances:

- You write replicated data to a database target and use a table name prefix.
  You can append a prefix to the names of database target tables. A table name prefix prevents you from overwriting database tables when you share a database account.

- You replicate case-sensitive data.
  When the Data Replication service replicates data to a target database, it creates all table and column names in uppercase. If the target database is case sensitive, use uppercase table and column names when you query the database.

- You replicate objects with long object and field names.
  When a source object or field name contains more characters than the maximum allowed for the name in the target, the Data Replication service truncates the table or column name in the target database.
Truncating Table Names

When you replicate a source object to a database, the Data Replication service replicates the data to a database table with the same name as the source object. If you replicate data to a database target and the length of the source object name exceeds the maximum number of characters allowed for the target table name, the Data Replication service truncates the table name in the target database. It truncates the table name to the first X characters, where X is the maximum number of characters allowed for a table name in the target database.

The following table lists the maximum number of characters allowed for a table name for each supported target database:

<table>
<thead>
<tr>
<th>Target Database</th>
<th>Maximum Number of Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server</td>
<td>64</td>
</tr>
<tr>
<td>MySQL</td>
<td>128</td>
</tr>
<tr>
<td>Oracle</td>
<td>30</td>
</tr>
</tbody>
</table>

Duplicate Tables Names from Same Data Replication Task

If you replicate multiple source objects from the same data replication task to the same database user account and truncation causes duplicate table names, the Data Replication service replaces the last character of the duplicate table names with sequential numbers.

For example, the data replication task contains the following Salesforce source objects:

- TenLetters1234567890TenLettersXXX
- TenLetters1234567890TenLettersYYY
- TenLetters1234567890TenLettersZZZ

When you replicate the objects, the Data Replication service creates the following truncated table names in the target database:

- TenLetters1234567890TenLetters
- TenLetters1234567890TenLetter1
- TenLetters1234567890TenLetter2

Duplicate Table Names from Different Data Replication Tasks

If you replicate multiple source objects with the same names from different data replication tasks to the same database user account, the Data Replication service creates one target table, and overwrites the table data each time you replicate one of the objects. If you run a full load, it overwrites the entire table. If you run an incremental load, it overwrites the changed rows.

To avoid overwriting tables, use a different target table name prefix for each data replication task.

RELATED TOPICS:
- “Target Prefixes” on page 95

Truncating Column Names

If the length of the source field name exceeds the maximum number of characters allowed for a column name in a relational target, the Data Replication service truncates the column name in the target database. It truncates the column name to the first X characters, where X is the maximum number of characters allowed for a column name in the target database.
The following table lists the maximum number of characters allowed for a column name for each supported target database:

<table>
<thead>
<tr>
<th>Target Database</th>
<th>Maximum Number of Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft SQL Server</td>
<td>64</td>
</tr>
<tr>
<td>MySQL</td>
<td>128</td>
</tr>
<tr>
<td>Oracle</td>
<td>30</td>
</tr>
</tbody>
</table>

For example, the Data Replication service creates a column name in an Oracle database based on the following 40-character field name of a Salesforce object:

```
TenLetters1234567890TenLettersXXXXXXXXXX
```

The Data Replication service truncates the column name to the first 30 characters:

```
TenLetters1234567890TenLetters
```

If the truncation causes duplicate column names for the target table, the Data Replication service replaces the last character of the duplicate column names with sequential numbers. The Data Replication service also replaces the last character of duplicate table names from the same data replication task.

## Resetting a Database Target

Reset a relational target table in a data replication task to drop the target table. You may want to drop the target table to re-create it based on the latest source object definition. When you reset a target table for a data replication task, the Data Replication service completes the following steps:

1. Drops the target table included in the data replication task from the database.
2. Sets the load type for the data replication task to full load. You must run a full load to reload the data from all source objects included in the data replication task.

When you run the data replication task after you reset the target, the Data Replication service recreates each target table and then loads all of the data into the new table.

You may need to reset a relational target for the following reasons:

- The datatype of a Salesforce source field changed and you run the data replication task configured with an incremental load.
  
  If the datatype of a source field changed and you run the data replication task configured with an incremental load, the data replication task fails because of a mismatch between the datatypes of the source field and the column of the relational target table. Reset the target table to synchronize the datatype of the target table column with the Salesforce field.

- The precision or scale of a Salesforce source field increases and you run the data replication task configured with a load.

  If the precision or scale of a Salesforce field increased and you run the data replication task configured for an incremental load, the data replication task fails because of the target table column may not be able to store all values from the Salesforce field. Reset the target table to synchronize the precision or scale of the target table column with the Salesforce field. If the precision or scale decreases, the data replication task succeeds and the target table columns remain unchanged.
You delete a field in a Salesforce source or a column in a database target and the data replication task writes to a Microsoft SQL Server relational target.

If you run a data replication task that writes to a Microsoft SQL Server target and the source is missing a field or column, the data replication task fails. To run the data replication task successfully, reset the target table to re-create the target based on the latest source definition, and then run the data replication task.

The relational target table is corrupt.

If the target table is corrupt, the data replication task may consistently fail to write to the target table.

To reset the relational target tables included in a data replication task:

1. Click Data Services > Data Replication.
2. Click Reset Target next to the applicable data replication task.

Rules and Guidelines for Resetting a Target Table

Use the following rules and guidelines when you reset a target table:

- If you previously created indexes on a target table and reset the target table, the Data Replication service drops the indexes and the target table. You must create the indexes again.
- An error appears when you try to reset a target table that does not exist in the target database.
- The Data Replication service drops the target table that was updated the last time the data replication task ran. For example, if you change the prefix of the target table and do not run the data replication task, the Data Replication service resets the old target table.

Creating Target Tables

You can use Informatica Cloud to create the database table for a target before you run the data replication task. You might want to create the target table, and then modify the table properties before the data replication task loads the data into the table. For example, you may want to add extra columns to the table or create indexes on particular columns.

To create the target table for a data replication task:

1. Click Data Services > Data Replication.
2. Click the Create Target option next to the applicable data replication task.

Understanding Load Types

Use one of the following load types when you replicate data:

- **Full load**
  The Data Replication service replicates all rows of the source objects in the data replication task.

- **Incremental load**
  The first time the Data Replication service runs an incremental load it performs a full load, replicating all rows of the source objects in the data replication task. For each subsequent run, the Data Replication service replicates rows that have changed since the last time it ran the task.
Full Load

For a full load, the Data Replication service replicates the data for all rows of the source objects in the data replication task. Each time the data replication task runs, the Data Replication service truncates the target database tables, or flat file, and performs a full data refresh from the source.

Run a full load in the following situations:

- The datatype of a Salesforce field changed.
  If the Data Replication service detects a datatype change, it will not run an incremental load. Reset the target table to create a table that matches the updated Salesforce object, and then run the data replication task with full load to reload the data for all Salesforce objects included in the data replication task.

- A Salesforce object in the data replication task is configured to be non-replicateable within Salesforce.
  If you run an incremental load on a data replication task that contains non-replicateable objects, the Data Replication service runs a full load on the object. Contact the Salesforce administrator to get a list of replicateable Salesforce objects.

Incremental Load

For an incremental load, the Data Replication service replicates the data for new rows and the rows that have changed since the last run of the data replication task. The time of the last run is determined by the time that the last record is replicated from Salesforce.

The first time you run a data replication task as an incremental load, the Data Replication service replicates data from all the rows, just like a full load. Each time the data replication task runs, the Data Replication service inserts new rows and updates changed rows. It does not update the unchanged rows already present in target tables.

When you run an incremental load, each replication session occurs in a single transaction. If errors occur, the entire transaction rolls back. To improve performance, avoid using an incremental load if you schedule the data replication task to run at long intervals, such as weeks or months. Instead, run a full load to avoid rollback of all data that was replicated during the schedule interval.

When the Data Replication service compares Salesforce source objects and target tables to find inconsistencies, it completes the following tasks:

1. Compares field names in the source and target.
2. Inserts and deletes fields in the target table to match the source.
3. Runs a query to determine if values have been inserted or updated.
4. Replicates new and changed rows.

When you run an incremental load, inconsistencies between the Salesforce object and target table affect the data that the Data Replication service replicates.

Rules and Guidelines for Running Incremental Loads

Use the following rules and guidelines when you run a data replication task as an incremental load:

- You can add a field to a target. Because the Data Replication service reconciles field name inconsistencies before it runs the query to find changed data, changes to field names do not necessarily cause rows to be replicated. Replication occurs only if the row data changes.
  For example, the Salesforce source contains a field named New that does not exist in the target. If you run an incremental load and no value exists in row 1 for the New field, the Data Replication service adds the New field to the target table, but it does not replicate the data in row 1. If the New field contains a value in row 1, then the Data Replication service replicates the data in row 1.
- The incremental load may fail if the datatype, precision, or scale for the source field and target column are inconsistent. The incremental load fails when the source and target columns have inconsistent datatypes or the source field has a higher precision or scale than the target column. To resolve the problem, reset the target to re-create the target tables to match the corresponding source objects. Next, run the data replication task with full load to reload the data for all source objects included in the data replication task.

- The Data Replication service cannot run an incremental load on a Salesforce object that is non-replicateable or that does not have Created By and Last Modified By dates. Each Salesforce object can be configured to be non-replicateable in Salesforce. Contact the Salesforce administrator to get a list of replicateable Salesforce objects. If you run a data replication task as an incremental load, the Data Replication service runs a full load on all non-replicateable objects and objects that do not have Created By and Last Modified By dates. Salesforce does not track the Created By or Last Modified By dates for all objects.

- The Data Replication service runs an incremental load as a full load when you run the data replication task for the first time. It also runs an incremental load as a full load the first time you run the task after changing any of the following task properties:
  - Source or target connection used in the task.
  - Properties of the source or target connection used in the task.
  - Load type of the task.

- You cannot run incremental loads on data replication tasks that have a database source.

- If you run a task with an incremental load multiple times, the time period between the end time of the previous run and the start time of the next run must be at least 60 seconds. The task fails if you try to run the task before the 60-second waiting period.
CHAPTER 7

Data Synchronization Service

This chapter includes the following topics:

- Data Synchronization Service Overview, 108
- Data Synchronization Prerequisite Tasks, 109
- Understanding Data Synchronization Task Options, 109
- Configuring a Data Synchronization Task, 117
- Promoting Data Loader Tasks to Data Synchronization Tasks, 128
- Viewing Data Synchronization Task Details, 129
- Copying a Data Synchronization Task, 129
- Running a Data Synchronization Task, 129
- Stopping a Data Synchronization Task, 130
- Monitoring a Data Synchronization Task, 131
- Downloading Mapping XML, 131
- Deleting a Data Synchronization Task, 131

Data Synchronization Service Overview

The Data Synchronization service allows you to synchronize data between a source and target. For example, you can read sales leads from your sales database and write them into Salesforce. You can also use expressions to transform the data according to your business logic or use data filters to filter data before writing it to targets.

Complete the prerequisite tasks before you create a data synchronization task. When you create a task, you can associate it with a schedule to run it at specified times or on regular intervals. Or, you can run it manually. You can monitor tasks that are currently running in the activity monitor and view logs about completed tasks in the activity log.

Use the Data Synchronization Task wizard to create a data synchronization task. The Data Synchronization Task wizard provides two modes:

- **Basic**
  Displays a full list of available objects when you select a connection. Displays preview data when you select a source or target object.

- **Advanced**
  Allows you to enter a source or target name and to choose to preview data. Use Advanced mode when a connection you want to use accesses a large number of source or target objects, or when source or target objects include a large volume of data.
Data Synchronization Prerequisite Tasks

Before you create a data synchronization task, complete the following prerequisite tasks:

1. Create the flat file.
   You must create the flat file and its directory before you can create a data synchronization task that reads from or writes to a flat file.

2. Create the target database.
   To write Salesforce data to a database target, the database administrator must create a database table to store the data for each Salesforce object. The database must meet the minimum system requirements.

3. Create database users.
   To write Salesforce data to a database target, the database administrator must create a database user account in the target database. Each database user account must have the DELETE, INSERT, SELECT, and UPDATE privileges.

4. Verify that the sources and targets meet the requirements.

5. Download and run a Secure Agent.
   If you read data from or write data to a flat file or database, download and run a Secure Agent on a machine that runs the data synchronization task.

Understanding Data Synchronization Task Options

The Data Synchronization Task wizard provides many options for you to configure tasks. Available options depend on the options that you select, such as the source and target connection or task operation. Advanced options allow you to perform tasks such as creating a target file or configuring post-processing commands.

This section describes the options you can use when you configure a data synchronization task.

Understanding Sources in a Data Synchronization Task

You can use a single file, database table, Salesforce object, or custom source object in a data synchronization task. You can also use up to five related database tables or five related Salesforce objects as sources for a data synchronization task.

Database tables are related when they have columns on which they can be joined. When you add multiple database tables as a source for a task, you can either define relationships with key columns or create user-defined join conditions.

Salesforce objects are related when they have an explicit relationship defined between them in Salesforce. For example, an Opportunity object has predefined relationships with the Account and Owner objects.

Including Historical Salesforce Data in a Task

When you use Salesforce sources in a task, you can include deleted or archived source data with the Include Archived and Deleted Rows in the Source option. When you select this source option with an insert or update task operation, the Data Synchronization service inserts deleted or archived Salesforce data to the target. When you use the delete task operation, the Data Synchronization service ignores deleted or archived data.
Rules and Guidelines for a Multiple-Object Source

Use the following rules and guidelines when configuring a multiple-object database or Salesforce source:

- A task with multiple Salesforce source objects and a flat file target fails if the Data Synchronization service cannot replicate all source objects when you run the task.
- All objects must be available through the same source connection.
  - All Salesforce objects in a multiple-object source must have a predefined relationship in Salesforce.
  - All database tables in a multiple-object source must have valid relationships defined by key columns or user-defined join conditions.
- When you add multiple database tables as sources for a task, you can either create relationships or user-defined joins, but not both.
- The Data Synchronization Task wizard removes a user-defined join under the following conditions:
  - You remove one of two remaining database tables from the list of sources for the task.
  - You change the source connection from database to flat file or Salesforce.
- Microsoft Access and ODBC source connections do not detect primary-foreign-key relationships. Therefore, these tables do not appear related when you add multiple source objects.
- If you add multiple MySQL tables as the source for a task and you use an ODBC connection to connect to MySQL, you must use a user-defined join. If you create the relationship and run the task, the task fails.

Understanding Targets in a Data Synchronization Task

You can use a file, database table, or Salesforce object as a target for a data synchronization task.

The target connections that you can use depend on the task operation you select for the task. For example, if you select the upsert task operation, you cannot use a flat file target connection because you cannot upsert records into a flat file target.

Creating a Flat File Target

If a task has a flat file target, you must create the flat file before you can save the task. You can create the flat file target with the Data Synchronization Task wizard when all of the following are true:

- The source connection type is Salesforce, database, or ODBC.
- The source object is Single or Custom.
- The target connection type is Flat File.

The Data Synchronization Task wizard uses the source object name as the default name of the flat file target. It truncates the name of the flat file to the first 100 characters if the source name is too long. If the target name conflicts with the name of another target object, the following error appears:

Object named <object name> already exists in the target connection.

Truncating Targets

You can configure a data synchronization task to truncate a database target table before writing new data to the table. You can truncate a target when you configure the task to use an Insert task operation and to write to a database target. By default, Informatica Cloud inserts new rows into without truncating the target table.

When Informatica Cloud truncates a target through an ODBC connection, it executes a DELETE FROM statement to truncate the table. When Informatica Cloud truncates a target using other connection types, it executes a TRUNCATE TABLE statement.
Rules and Guidelines for Data Synchronization Sources and Targets

Use the following rules and guidelines for data synchronization sources and targets:

- Field names must contain 65 characters or less.
- Field names must contain only alphanumeric or underscore characters. Spaces are not allowed.
- Field names cannot start with a number.
- Each field name must be unique within each source and target object.
- The service truncates data if the scale or precision of a numeric target column is less than the scale or precision of the corresponding source column.

Rules and Guidelines for Flat File Sources and Targets

Use the following rules and guidelines for flat file sources and targets:

- All date columns in a flat file source must have the same date format. Rows that have dates in a different format than the one specified in the data synchronization task definition are written to the error log file.
- Each flat file target must contain all fields that will be populated by the data synchronization task.
- The flat file cannot contain empty column names. If a file contains an empty column name, the following error appears:
  
  Invalid header line: Empty column name found.

- Column names in a flat file must contain printable tab or ASCII characters (ASCII code 32 - 126). If the file contains an invalid character, the following error appears:
  
  Invalid header line: Non-printable character found. The file might be binary or might have invalid characters in the header line.

- If the column name of a flat file contains non-alphanumeric characters, starts with a number, or contains more than 75 characters, the Data Synchronization service modifies the column name. The Data Synchronization service truncates column names to 75 characters. For a flat file source, the data preview and expression dialog box shows modified column names. For a flat file target, the Data Synchronization service changes the column name in the flat file when it generates the file at runtime.

Rules and Guidelines for Database Sources and Targets

Use the following rules and guidelines for database sources and targets:

- You can use database tables as targets. You can use database tables, aliases, and views as sources. You can also use an Oracle synonym as a source.
- Relational targets must meet the minimum system requirements.
- The data synchronization task fails under the following conditions:
  - The task writes to an Oracle target field of a blob, bfile, or raw datatype. You cannot write source data to Oracle target fields with binary datatypes.
  - The task has an upsert, an update, or a delete task operation and contains a Microsoft Access target that uses an MS Access connection.
  - The task contains a lookup and a Microsoft Access target that contains a nullable primary key field.
- If a task writes time data from MySQL to a flat file, the Data Synchronization service converts the time data to a Date/Time datatype, where the date is the current date and time is the time specified in the source. You can use a string function in an expression to remove the date before loading to the flat file.
- While reading time data from a MySQL source, the Data Synchronization service converts the time to the Coordinated Universal Time (UTC) time zone. Consider the time zone change if you create data filters and field mapping expressions based on time.
The database user account for each database target connection must have the DELETE, INSERT, SELECT, and UPDATE privileges.

Rules and Guidelines for Salesforce Sources and Targets

Use the following rules and guidelines for Salesforce sources and targets:

- When you create a connection to a Salesforce source or target, the Informatica Cloud caches the connection metadata. If the connection metadata changes while you are creating the connection, you must log out and log back in to refresh the connection metadata.
- The source must provide non-null values for required fields in a Salesforce target object.
- When you use the task operation for a Salesforce object target, specify the external ID for each related object.
- When you write Salesforce data to a database target, verify that the Salesforce data uses the following required formats for date and time datatypes: Date (yyyy-MM-dd) and DateTime (yyyy-MM-dd HH:mm:ss).
  If a record contains the date and time in a different format, the Data Synchronization service writes the row to the error log file. If the Salesforce source contains a date and no time for the datetime datatype, the Data Synchronization service appends '00:00:00' at the end of the date value to ensure the value is in the required format. When you write to a database target, the Data Synchronization service converts the Salesforce date and datetime data to the correct format expected by the database.
- A task may lose the least significant portion of numerical data in a Salesforce target field when the data uses most of the maximum precision and scale specified for the datatype of the field.
  For example, when trying to insert 65656656565656.6969 into a Salesforce field with datatype Decimal(14, 4), the task inserts 65,656,565,656,565.6950 instead. Or, when trying to insert 123456789123456789 into a Salesforce field with datatype Decimal(18, 0), the task inserts 123,456,789,123,456,784.
- A task may round data unexpectedly into a Salesforce target field when the data uses most of the maximum precision and scale specified for the datatype of the field.
  For example, when trying to insert 555555555565.8855 into a Salesforce field with datatype Decimal(18, 0), the task inserts 555555555566 instead. However, when you manually enter the data in Salesforce, Salesforce rounds the data as 555555555565. Or, when trying to insert 12345678923456.3399 into a Salesforce field with datatype Decimal(14, 4), the task inserts 12,345,678,923,456.3400 instead.

Understanding Field Mappings in a Data Synchronization Task

Map source columns to target columns on the Field Mapping page of the Data Synchronization Task wizard. You must map at least one source column to a target column. Map columns with compatible datatypes or create field expressions to convert datatypes appropriately.

Depending on the task operation, the Data Synchronization service requires certain fields to be included in the field mapping. By default, the Data Synchronization service maps the required fields. If you configure the field mapping, ensure that the required fields remain mapped. If you do not map the required fields, the data synchronization task fails.

The following table shows the required fields for each applicable task operation and target type:

<table>
<thead>
<tr>
<th>Required Field</th>
<th>Task Operations</th>
<th>Target Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Delete Update</td>
<td>Salesforce</td>
<td>Map the ID column to enable the Data Synchronization service to identify records to delete or update in a Salesforce target.</td>
</tr>
</tbody>
</table>
### Required Field | Task Operations | Target Type | Description
--- | --- | --- | ---
Upsert | External ID | Upsert | Map at least one External ID field to enable the Data Synchronization service to identify records to upsert in a Salesforce target.
Primary Keys | Delete Update Upsert | Database | Map primary key columns to enable the Data Synchronization service to identify records to delete, update, or upsert in a database target.
Non-null fields | Insert Update Upsert | Salesforce | Map all fields that cannot be null in Salesforce.

When you configure field mappings, you can also perform the following tasks:
- Automatically map columns with similar names.
- Edit field datatypes.
- Add a plug-in to the field mapping.
- Create field expressions to transform data.
- Create lookups.
- Validate field mappings.

**RELATED TOPICS:**
- “Rules and Guidelines for Validating Field Mappings” on page 141
- “Field Expressions” on page 141

### Field Datatypes
When you create a data synchronization task, Informatica Cloud assigns a datatype to each field in the source and target. You can configure the datatypes of fields in sources or targets with flat file, Microsoft Access, ODBC, or FTP/SFTP connection types.

### Plug-ins in Field Mappings
You can add a plug-in to a field mapping. After you add a plug-in to a field mapping, you must map the source fields to the input fields of the plug-in and map the output fields of the plug-in to the target fields.

When a source field is mapped directly to a target field and you map an output field of a plug-in to the same target field, Informatica Cloud concatenates the values of the source and output fields in the target field. Verify that the expression is correct for the target field.

**Note:** The names of the output fields of a plug-in do not match the source field names. Informatica Cloud appends a number to the end of the source field name to determine the output field name. In addition, Informatica Cloud may not display the output fields in the same order as the source fields.
Lookups

You can create a lookup to return information from an object in any Salesforce, database, or flat file connection. For example, if you use the Salesforce Account object as a source, you can look up the name of the owner in the Salesforce Contact object.

The lookup returns values based on a lookup condition. You create a lookup condition based on information in the source. For example, for a SALES source database table, you might set the ITEM_ID column equal to ITEM_ID column in a ITEMS flat file, and have the lookup return the item name for each matching item ID.

When you create a lookup condition, you define the following components:

- **Lookup connection and object.** The connection and object to use to perform the lookup.
- **Lookup fields.** Source and lookup fields used to define the lookup condition. The service compares the value of the source field against the lookup field, and then returns the output value based on the match. You can define multiple conditions in a lookup. If you define more than one lookup condition, all lookup conditions must be true to find the match.

For example, you define the following conditions for a lookup:

```
SourceTable.Name = LookupTable.Name
SourceTable.ID = LookupTable.ID
```

The service performs the following lookup:

```
Lookup (SourceTable.Name = LookupTable.Name, SourceTable.ID = LookupTable.ID)
```

- **Lookup return value.** The value that the task returns when it finds a match. You must also configure how the service handles multiple return values. The service can randomly choose one of the return values or return an error. When the lookup returns an error, the service writes the row to the error log file.

Rules and Guidelines for Lookups

Use the following rules and guidelines when creating a lookup:

- If the lookup is on a flat file, the file must use a comma delimiter. You cannot use any other type of delimiter.
- Tasks with a flat file lookup that run by a Secure Agent on Windows 7 (64 bit) might not complete. To resolve the issue, configure a network login for the Secure Agent service.
- For each source field, you can perform a lookup or create an expression. You cannot do both.
- Each task can contain one or more lookups on Salesforce objects, database tables, and flat files. To avoid impacting performance, include less than six lookups in a task.
- When performing the lookup, the service performs an outerjoin and does not sort the input rows. The lookup performs a string comparison that is not case-sensitive to determine matching rows.
- The source field and lookup field in the lookup condition must have compatible datatypes. If the datatypes are not compatible, the following error appears:

```
Source field [<source field name> (<source field datatype>)] and lookup field [<lookup field name> (<lookup field datatype>)] have incompatible datatypes.
```

If you create multiple lookup conditions on a lookup field and the lookup source is a flat file, all source fields must have the same datatype. The Data Synchronization service uses the larger precision and scale of the source field datatypes as the precision and scale for the target field. If the source fields do not have the same datatype, the following error appears:

```
Lookup field <field name> in <file name> has conflict data types inferred: <datatype 1> and <datatype 2>.
```
You cannot include lookup fields of particular datatypes in a lookup condition. The task fails in the following cases:
- The lookup field in a MySQL database table has the Text or Ntext datatype, and the MySQL connection has a UTF-8 code page.
- The lookup field in a Salesforce object has the Text Area (Long) datatype.
- The lookup field in a flat file has the Text or Ntext datatype or the target field of a lookup has the Text or Ntext datatype.
- If you run a task with a lookup and the source field, lookup field, or output field of the lookup no longer exist in the lookup object, an error appears.

Running or Scheduling a Data Synchronization Task

You can configure a task to run on demand or to run on a schedule. When you configure how to run a task, you can also configure the following options:
- Email notification
- Preprocessing commands
- Postprocessing commands
- Update columns
- Advanced Salesforce target options

Understanding Email Notification Options

You can configure email notification for a Data Synchronization or Data Replication task. When you configure email notification for the task, Informatica Cloud uses the email notification options configured for the task instead of the email notification options configured for the organization. You can send email to different addresses based on the status of the task:
- Success. The task completed successfully.
- Warning. The task completed with errors.
- Error. The task did not complete.

Understanding Preprocessing and Postprocessing Commands

You can run preprocessing and postprocessing commands to perform additional tasks. The service runs preprocessing commands before it reads the source. It runs postprocessing commands after it writes to the target.

You can use the following types of commands:
- SQL commands. Use SQL commands to perform database tasks.
- Operating system commands. Use shell and DOS commands to perform operating system tasks.

If any command in the preprocessing or postprocessing scripts fail, the service fails the task.

Rules and Guidelines for Preprocessing and Postprocessing SQL Commands

You can run SQL commands before or after a task. For example, you can use SQL commands to drop indexes on the target before the task runs, and then recreate them when the task completes.

Use the following rules and guidelines when creating the SQL commands:
- Use any command that is valid for the database type. However, Informatica Cloud does not allow nested comments, even if the database allows them.
• Use a semicolon (;) to separate multiple statements. Informatica Cloud issues a commit after each statement.
• Informatica Cloud ignores semicolons within /* ...*/.
• If you need to use a semicolon outside of comments, you can escape it with a backslash (\).
• Informatica Cloud does not validate the SQL.

Rules and Guidelines for Preprocessing and Postprocessing Operating System Commands
Informatica Cloud can perform operating system commands before or after the task runs. For example, use a preprocessing shell command to archive a copy of the target flat file before the task runs on a UNIX machine.

You can use the following types of operating system commands:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Command Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX</td>
<td>Any valid UNIX command or shell script.</td>
</tr>
<tr>
<td>Windows</td>
<td>Any valid DOS or batch file.</td>
</tr>
</tbody>
</table>

Update Columns
Update columns are columns that uniquely identify rows in the target table. Add update columns when the database target table does not contain a primary key, and the data synchronization task uses an update, upsert, or delete task operation.

When you run the data synchronization task, the Data Synchronization service uses the field mapping to match rows in the source to the database table. If the Data Synchronization service matches a source row to multiple target rows, it performs the specified task operation on all matched target rows.

Salesforce Success Files
The Salesforce success file contains one row for each successfully processed row. Each row contains the row ID, data, and one of the following task operations: Created, Updated, Upserted, or Deleted.

Use the success file to track rows that are created in case you need to roll back the operation. The Data Synchronization service stores the success files in the directory where you installed the Secure Agent:

<SecureAgent_InstallDirectory>\main\rtdmDir\success

Salesforce Target Batch Size
The target batch size determines the maximum number of records to include in each query that writes to the Salesforce target. Salesforce allows up to 200 records for each query. If you enter a value higher than 200, only 200 rows are included in each query. Default is 200.

You might use a smaller batch size for upserts because you cannot update the same row more than once in a single query. To process multiple upserts on a particular row in the same query, set the batch size to 1.

Salesforce limits the number of queries you can make in a 24-hour period. For more information about the limit, see the Salesforce documentation.
Configuring a Data Synchronization Task

You can configure a data synchronization task using the Data Synchronization Task Wizard.

To configure a data synchronization task using the Data Synchronization Task Wizard, complete the following steps:

1. Define the data synchronization task.
2. Configure the source.
3. Configure the target.
4. Configure the data filters.
5. Configure field mappings.
6. Configure the schedule.
   If applicable, you can also configure the advanced options for a data synchronization task that writes to a Salesforce target.

The procedure for each step depends on whether you use the Data Synchronization Wizard in basic or advanced mode.

**Related Topics:**
- “Understanding Data Synchronization Task Options” on page 109

**Step 1. Define the Data Synchronization Task**

Define the data synchronization task.

1. To define a data synchronization task, click **Data Services > Data Synchronization**, and then click **New**.
   To edit the definition of an existing data synchronization task, click **Data Services > Data Synchronization**, click the name of the data synchronization task, and then click **Edit**.

2. In the Data Synchronization Task Details area, configure the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Task Name   | Name of the data synchronization task. The names of data synchronization tasks must be unique within the organization. Data synchronization task names can contain alphanumeric characters, spaces, and the following special characters: _ . + -
Data synchronization task names are not case sensitive. |
| Description | Description of the data synchronization task. Maximum length is 255 characters. |
| Task Operation | Select one of the following task operation types:
- Insert
- Update
- Upsert
- Delete |
The list of available targets in a subsequent step depend on the operation you select.

Wizard Mode

Select the mode the Data Synchronization wizard uses to create or edit the task.
- Select Basic to display all available objects when you select a connection and to preview data by default.
- Select Advanced to select or enter a source name and to preview data on demand. Use when a connection contains a large number of object or when source or target objects contain a large volume of data.

3. Click **Next**.

### Step 2. Configure the Source

Select the source for the data synchronization task. The steps to configure a source depend on the wizard mode and source object type you select.

The following table describes the procedures you can use:

<table>
<thead>
<tr>
<th>Source Type</th>
<th>Wizard Mode</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single object or custom source</td>
<td>Basic</td>
<td>&quot;Configuring a Single Object or Custom Source as the Source (Basic)&quot; on page 118.</td>
</tr>
<tr>
<td>Single object or custom source</td>
<td>Advanced</td>
<td>&quot;Configuring a Single Object or Custom Source as the Source (Advanced)&quot; on page 119.</td>
</tr>
<tr>
<td>Multiple database tables</td>
<td>Basic, Advanced</td>
<td>&quot;Configuring Multiple Database Tables as the Source&quot; on page 120.</td>
</tr>
<tr>
<td>Multiple Salesforce objects</td>
<td>Basic, Advanced</td>
<td>&quot;Configuring Multiple Salesforce Objects as the Source&quot; on page 121.</td>
</tr>
</tbody>
</table>

**RELATED TOPICS:**
- “Understanding Data Synchronization Task Options” on page 109

**Configuring a Single Object or Custom Source as the Source (Basic)**

You can configure a single object as the source of a data synchronization task. You can also configure a custom source as the source of a data synchronization task.

1. To configure a single object or custom source for a new data synchronization task, click **Data Services** > **Data Synchronization** > **New**, complete the preliminary wizard steps, and then click **Source**.
   To configure a single object or custom source for an existing data synchronization task, click **Data Services** > **Data Synchronization**, click the name of the data synchronization task, click **Edit**, and then click **Source**.
2. On the Source page, select a connection.
   Click **New** to create a connection or click **Edit** to edit a connection.
3. To use a single source, select **Single** and select a source.
   To use a custom source, select **Custom** and select a source.
   If you have created a custom source, you can select a custom source when you use a database connection.
   When you select a source, the Data Preview area shows the first ten rows of the first five columns in the source. It also displays the total number of columns in the source.
If the source data contains binary data, the Data Preview area shows the following text:

BINARY DATA

4. For a Flat File or FTP/SFTP single source, select one of the following delimiters for a source flat file:
   - Comma
   - Tab
   - Colon
   - Semicolon
   - Other

If you choose Other, the delimiter cannot be an alphanumeric character or a double quotation mark. If you choose a delimiter for an FTP/SFTP flat file, Informatica Cloud applies the delimiter to the local file, not the remote file, when previewing and reading data. If the remote and local files are not synchronized, Informatica Cloud might produce unexpected results.

5. For Salesforce sources, if you want the Data Synchronization service to read historical data in Salesforce sources, including archived and deleted rows, select Include Archived and Deleted Rows in the Source.
   
By default, the Data Synchronization service reads only current source rows in selected Salesforce sources and ignores archived and deleted rows.

6. To preview all source columns in a file, click Preview All Columns.
   The file shows the first ten rows of the source.

7. Click Next.

Configuring a Single Object or Custom Source as the Source (Advanced)

You can configure a single object as the source of a data synchronization task. You can also configure a custom source as the source of a data synchronization task.

1. To configure a single object or custom source for a new data synchronization task, click Data Services > Data Synchronization > New, complete the preliminary wizard steps, and then click Source.
   
To configure a single object or custom source for an existing data synchronization task, click Data Services > Data Synchronization, click the name of the data synchronization task, click Edit, and then click Source.

2. On the Source page, select a connection.
   
Click New to create a connection or click Edit to edit a connection.

3. To use a single source, select Single, and then click Select.
   
To use a custom source, select Custom, and click Select.
   
If you have created a custom source, you can select a custom source when you use a database connection.

4. In the Select Source Object dialog box, select the source you want to use and click OK.
   You can also type the name of the source in the Selected Object field.

5. For a Flat File or FTP/SFTP single source, select one of the following delimiters for a source flat file:
   - Comma
   - Tab
   - Colon
   - Semicolon
   - Other

If you choose Other, the delimiter cannot be an alphanumeric character or a double quotation mark. If you choose a delimiter for a FTP/SFTP flat file, Informatica Cloud applies the delimiter to the local file, not the
remote file, when previewing and reading data. If the remote and local files are not synchronized, Informatica Cloud might produce unexpected results.

6. For Salesforce sources, if you want the Data Synchronization service to read historical data in Salesforce sources, including archived and deleted rows, select Include Archived and Deleted Rows in the Source.

   By default, the Data Synchronization service reads only current source rows in selected Salesforce sources and ignores archived and deleted rows.

7. To preview source data, click Show Data Preview.

   The Data Preview area shows the first ten rows of the first five columns in the source. It also displays the total number of columns in the source.

   If the source data contains binary data, the Data Preview area shows the following text:

   **BINARY DATA**

8. To preview all source columns in a file, click Preview All Columns.

   The file shows the first ten rows of the source.

9. Click Next.

**Configuring Multiple Database Tables as the Source**

You can configure multiple database tables as the source of a data synchronization task.

1. To configure a multiple database table source for a new data synchronization task, click Data Services > Data Synchronization > New, complete the preliminary wizard steps, and then click Source.

   To configure a multiple database table source for an existing data synchronization task, click Data Services > Data Synchronization, click the name of the data synchronization task, click Edit, and then click Source.

2. Select a database connection.

   Click New to create a connection or click Edit to edit a connection.

3. Select Multiple.

   The Source Objects table displays. The Action column lists the actions that you can perform on each row. The Source Object column shows each selected source object. The Join Condition column shows the join conditions for the database table.

4. Click Add.

5. In the Select Objects dialog box, select up to 5 objects, and then click OK.

   The selected sources display in the Source Objects table. To remove a source, click the Remove icon in Action column for the source.

6. To add additional sources related to a selected source, highlight a source in the Source Objects table and click Add.

7. In the Select Objects dialog box, select the sources you want to use, and then click OK.

8. Create source relationships or create a user-defined join, and click OK.

   - Create Relationship
     Select a database table and click Create Relationship. In the Create Relationships dialog box, select the source key for the table. Select the related source and matching object key, and then click OK. Match the primary key of the source table to the corresponding foreign key of the related database table. If you create more than one relationship for the same table, the Source Objects table shows each relationship in a different row of the table.
To join all database tables, select **User Defined Join**. When you select this option, the Data Synchronization Task Wizard adds existing relationships to the join condition. To ensure you enter field names correctly, you can select an object and the associated field to add it to the join statement.

9. To preview source data in basic mode, select the source in the Source Objects table.
   
   To preview source data in advanced mode, select the source in the Source Objects table and click **Show Data Preview**.
   
   The Data Preview area shows the first ten rows of the first five columns in the source. It also displays the total number of columns in the source.
   
   If the source data contains binary data, the Data Preview area shows the following text:
   
   BINARY DATA

10. To preview all source columns in a file, select the source in the Source Objects table and click **Preview All Columns**.
    
    The file shows the first ten rows of the source.

11. Click **Next**.

### Configuring Multiple Salesforce Objects as the Source

You can configure multiple Salesforce objects as the source of a data synchronization task.

1. To configure a multiple-object source for a new data synchronization task, click **Data Services > Data Synchronization > New**, complete the preliminary wizard steps, and then click **Source**.
   
   To configure a multiple-object source for an existing data synchronization task, click **Data Services > Data Synchronization**, click the name of the data synchronization task, click **Edit**, and then click **Source**.

2. Select the Salesforce connection.
   
   Click **New** to create a connection or click **Edit** to edit a connection.

3. Select **Multiple**.
   
   The Source Objects table displays. The Action column lists the actions that you can perform on each row. The Source Object column shows each selected Salesforce source. The Relationship Path column shows the relationship between the source and the primary Salesforce source.

4. Click **Add** to add the primary Salesforce source.

5. In the Add Object dialog box, select the primary Salesforce source you want to use, and then click **OK**.

6. To add a source related to the primary source, highlight the primary source in the Source Objects table, and then click **Add**.
   
   The Add Related Objects dialog box displays. The dialog box shows the available relationships that have not been associated with the selected source object. The Data Synchronization Task Wizard selects the object based on the selected relationship.

7. Select a relationship type and click **OK**.
   
   The selected relationship path displays in the Source Objects table. To remove a source, click the Remove icon in Action column for the source.

8. To add additional related sources, select the source you want to use and click **OK**. Then select another relationship type and click **OK**.

9. If you want the Data Synchronization service to read historical data in Salesforce sources, including archived and deleted rows, select **Include Archived and Deleted Rows in the Source**.
   
   By default, the Data Synchronization service reads only current source rows in selected Salesforce sources and ignores archived and deleted rows.
10. To preview source data in basic mode, select the source in the Source Objects table.
   To preview source data in advanced mode, select the source in the Source Objects table and click **Show Data Preview**.
   The Data Preview area shows the first ten rows of the first five columns in the source. It also displays the total number of columns in the source.
   If the source data contains binary data, the Data Preview area shows the following text:
   
   **BINARY DATA**

11. To preview all source columns in a file, select the source in the Source Objects table and click **Preview All Columns**.
   The file shows the first ten rows of the source.

12. Click **Next**.

### Step 3. Configure the Target

The steps to configure a target depend on the Data Synchronization Task wizard mode you select. Available options depend on the source type and task operation that you select for the task.

The following table describes the procedures you can use:

<table>
<thead>
<tr>
<th>Wizard Mode</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>&quot;Configure the Target (Basic)&quot; on page 122.</td>
</tr>
<tr>
<td>Advanced</td>
<td>&quot;Configure the Target (Advanced)&quot; on page 123.</td>
</tr>
</tbody>
</table>

**Related Topics:**
- "Understanding Targets in a Data Synchronization Task" on page 110

### Configure the Target (Basic)

Configure the target for the data synchronization task.

1. To configure a target for a new data synchronization task, click **Data Services > Data Synchronization > New**, complete the preliminary wizard steps, and then click **Target**.
   
   To configure a target for an existing data synchronization task, click **Data Services > Data Synchronization**, click the name of the data synchronization task, click **Edit**, and then click **Target**.

2. On the Target page, enter the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Select a connection. The list of available connections depends on the task operation defined for the task. Click New to create a connection or click Edit to edit a connection.</td>
</tr>
<tr>
<td>Create Target</td>
<td>Flat File connection only. Creates a target file. You can create a target file only when the source connection is Salesforce, database, or ODBC, and the source object is Single or Custom.</td>
</tr>
<tr>
<td>Target Object</td>
<td>Select the target object to which you want to write data.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Delimiter</td>
<td>For Flat File and FTP/SFTP connections only. Select one of the following delimiters for a target file:</td>
</tr>
<tr>
<td></td>
<td>- Comma</td>
</tr>
<tr>
<td></td>
<td>- Tab</td>
</tr>
<tr>
<td></td>
<td>- Colon</td>
</tr>
<tr>
<td></td>
<td>- Semicolon</td>
</tr>
<tr>
<td></td>
<td>- Other</td>
</tr>
<tr>
<td></td>
<td>If you choose Other, the delimiter cannot be an alphanumeric character or a double quotation mark. If you choose a delimiter for an FTP/SFTP flat file, Informatica Cloud applies the delimiter to the local file, not the remote file, when previewing and reading data. If the remote and local files are not synchronized, Informatica Cloud may produce unexpected results.</td>
</tr>
<tr>
<td>Truncate Target</td>
<td>Database targets with the Insert task operation only. Truncates a database target table before inserting new rows. Select True to truncate the target table before inserting all rows. Select False to insert new rows into existing target table without truncating the table. Default is False.</td>
</tr>
</tbody>
</table>

When you select a target, the Data Preview area shows the first ten rows of the first five columns in the target. It also displays the total number of columns in the target.

If the source data contains binary data, the Data Preview area shows the following text:

**BINARY DATA**

3. To preview all target columns in a file, click **Preview All Columns**.
   
   The file shows the first ten rows of the target.

4. If you do not want to create a target file, click **Next** to continue configuring the task.

5. To create a target file, click **Create Target**.

6. In Target File Details dialog box, enter the name of the file.

7. Click the right arrow to move the source fields you want to create to the Target Columns box.
   
   Use Shift-click to select contiguous fields. Use Ctrl-click to select non-contiguous fields.
   
   You can use the up and down arrows to order the target columns.

8. Click **OK**.

9. Click **Next**.

**Configure the Target (Advanced)**

Configure the target for the data synchronization task.

1. To configure a target for a new data synchronization task, click **Data Services > Data Synchronization > New**, complete the preliminary wizard steps, and then click **Target**.
   
   To configure a target for an existing data synchronization task, click **Data Services > Data Synchronization**, click the name of the data synchronization task, click **Edit**, and then click **Target**.

2. On the Target page, select a connection.
   
   The list of available connections depends on the task operation defined for the task.
   
   Click **New** to create a connection or click **Edit** to edit a connection.

3. To select a target, click **Select**.

4. In the Select Target Object dialog box, select the target you want to use and click **OK**.
   
   You can also type the name of the target in the Selected Object field.
5. Enter the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Target</td>
<td>Flat file connection only. Creates a target flat file. You can create a target file only when the source connection is Salesforce, database, or ODBC, and the source object is Single or Custom.</td>
</tr>
</tbody>
</table>
| Delimiter      | Flat File and Secure FTP connections only. Select one of the following delimiters for a target file:  
  - Comma  
  - Tab  
  - Colon  
  - Semicolon  
  - Other  
  If you choose Other, the delimiter cannot be an alphanumeric character or a double quotation mark. If you choose a delimiter for an FTP/SFTP flat file, Informatica Cloud applies the delimiter to the local file, not the remote file, when previewing and reading data. If the remote and local files are not synchronized, Informatica Cloud may produce unexpected results. |
| Truncate Target| Database targets with the Insert task operation only. Truncates a database target table before inserting new rows. Select True to truncate the target table before inserting all rows. Select False to insert new rows into existing target table without truncating the table. Default is False. |

6. To preview target data, click **Show Data Preview**.
   The Data Preview area shows the first ten rows of the first five columns in the target. It also displays the total number of columns in the target.
   If the source data contains binary data, the Data Preview area shows the following text:
   
   BINARY DATA

7. To preview all target columns in a file, click **Preview All Columns**.
   The file shows the first ten rows of the target.

8. If you do not want to create a target file, click **Next** to continue configuring the task.

9. To create a target file, click **Create Target**.

10. In Target File Details dialog box, enter the name of the file.

11. Click the right arrow to move the source fields you want to create to the Target Columns box.
    Use Shift-click to select contiguous fields. Use Ctrl-click to select non-contiguous fields.
    You can use the up and down arrows to order the target columns.

12. Click **OK**.

13. Click **Next**.

**Step 4. Configure the Data Filters**

Use a data filter to reduce the number of source rows that the Data Synchronization service reads for the task. By default, the Data Synchronization service reads all source rows.

1. To add data filters in a new data synchronization task, click **Data Services > Data Synchronization > New**, complete the preliminary wizard steps, and then click **Data Filters**.
   To add or change data filters in an existing data synchronization task, click **Data Services > Data Synchronization**, click the name of the data synchronization task, click **Edit**, and then click **Data Filters**.

2. To read all rows in a Salesforce source, select **No Limit**, or to read the first set of rows in a Salesforce source, select **Process Only the First** and enter a number.
3. To create a data filter, click **New**.

4. To create a simple data filter, on the Data Filters page, select a source column and operator and enter the value you want to use.

To create an advanced data filter, click **Advanced**. Enter the field expression you want to use and click **OK**. You can use the Delete icon to delete a data filter.

5. Click **Next**.

**RELATED TOPICS:**
- “Data Filters” on page 145

**Step 5. Configure the Field Mapping**

Configure field mappings to define the data the Data Synchronization service writes to the target.

1. To configure a field mapping in a new data synchronization task, click **Data Services > Data Synchronization > New**, complete the preliminary wizard steps, and then click **Field Mapping**.

   To configure a field mapping in an existing data synchronization task, click **Data Services > Data Synchronization**, click the name of the data synchronization task, click **Edit**, and then click **Field Mapping**.

2. If you included multiple source objects in the task, you can select each source object in the Sources field to show only the fields of that source object.

   By default, all source objects and their fields appear. Field names are grouped by source object. If you show all source objects, place the cursor over the status icon for a source field to determine the following information:

   - Datatype of a field.
   - Label and API names of Salesforce source fields. Label names appear in the Salesforce UI. API names are the names required by the Salesforce API.
   - Database table or Salesforce object to which the field belongs.

3. To add a plug-in, click **Add Plug-in**. In the **Add Plug-in** dialog box, select the plug-in.

   In the Mapping Selection area, select one of the following options and click **OK**:

   - **Source to Target**. Displays the source and target. Map source fields to the applicable target fields.
   - **Source to Plug-in**. Displays the source and the input fields of the plug-in. Map the source fields to the applicable input fields of the plug-in.
   - **Plug-in to Target**. Displays the output fields of the plug-in and the target fields. Map the output fields of the plug-in to the applicable target fields.

   If you select one of the above options, the Validate Mapping, Clear Mapping, and Auto-Match buttons only apply to the area of the field mapping that you select.

4. On the Field Mapping page, click **Auto-Match** to match fields with similar names.

   Or, you can select and drag the source fields to the applicable target fields.

   If the fields do not appear or do not appear correctly for a source or target, click **Refresh Fields**. If you change the source or target, Informatica Cloud does not automatically refresh the fields. Informatica Cloud caches the field metadata. Refresh the fields to update the cache and view the latest field attributes of a source or target.

   **Note:** Fields can be refreshed when the database server for the database source or target and the Secure Agent are all running.

5. To configure field datatypes for a Flat File, Microsoft Access, ODBC, or FTP/SFTP target, click **Edit Types**.
In the Edit Field Datatypes dialog box, configure the following datatype attributes and click **OK**:

<table>
<thead>
<tr>
<th>Datatype Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datatype</td>
<td>The datatype of data in the column.</td>
</tr>
<tr>
<td>Precision</td>
<td>Total number of digits in a number. For example, the number 123.45 has a precision of 5. The precision must be greater than or equal to 1.</td>
</tr>
<tr>
<td>Scale</td>
<td>Number of digits to the right of the decimal point of a number. For example, the number 123.45 has a scale of 2. Scale must be greater than or equal to 0. The scale of a number must be less than its precision. The maximum scale for a Numeric datatype is 65535.</td>
</tr>
</tbody>
</table>

6. To create an expression to transform data, click the **Add or Edit Expression** icon in the Actions column. In the Field Expressions dialog box, enter the expression you want to use and click **OK**.

7. To create a lookup, click the **Add or Edit Lookup** icon. In the Field Lookup dialog box, configure the following properties and click **OK**:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Connection for the lookup object.</td>
</tr>
<tr>
<td>Object</td>
<td>Object on which you want to lookup a value.</td>
</tr>
<tr>
<td>Source Fields</td>
<td>Source column to use in the lookup condition.</td>
</tr>
<tr>
<td>Lookup Fields</td>
<td>The column in the lookup table to use in the lookup condition.</td>
</tr>
<tr>
<td>Output Field</td>
<td>The column in the lookup table that contains the output value.</td>
</tr>
<tr>
<td>Multiplicity</td>
<td>Determines how to handle cases when a lookup returns multiple values: - Randomly Pick 1 Output Value. Select if the service should choose the first returned value when a lookup condition returns multiple values. Different source or target systems may not return lookup rows in the same order. - Error if More than One Output Value. Select if the service should display an error when the lookup condition returns multiple values. The service loads all rows for which only one match is found and rejects rows for which multiple matches are found. This is the default.</td>
</tr>
</tbody>
</table>

8. To configure external IDs for related Salesforce objects, click **Related Objects**. In the Related Objects dialog box, select the external ID for each applicable related object and click **OK**. You do not need to specify the external ID for a related object if you do not want to update changes for that related object. If you do not specify the external ID, the service requires that the source provide the Salesforce ID to uniquely identify records in each related object.

9. To clear an expression or lookup and delete the field mapping, click the **Clear Expression/Lookup** icon next to the target field.

10. To clear all field mappings, click **Clear Mapping**.

11. To validate a mapping, click **Validate Mapping**.

12. Click **Next**.
Step 6. Configure a Schedule

You can run a data synchronization task manually or schedule it to run at a specific time or on a time interval. You can create a schedule or use an existing schedule. You can also configure other advanced options for the task.

1. To add a schedule in a new data synchronization task, click Data Services > Data Synchronization > New, complete the preliminary wizard steps, and then click Schedule.

   To add or change a schedule in an existing data synchronization task, click Data Services > Data Synchronization, click the name of the data synchronization task, click Edit, and then click Schedule.

2. On the Schedule page, click Run this task on schedule and select the schedule you want to use.

   To create a new schedule, click New. Enter schedule details and click OK.

   To remove the task from a schedule, click Do not run this task on a schedule.

3. If the task includes Salesforce sources and targets, select a Secure Agent to run the task.

4. Configure email notification options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Default Email Notification Options for my Organization</td>
<td>Use the email notification options configured for the organization.</td>
</tr>
<tr>
<td>Use Custom Email Notification Options for this Task</td>
<td>Use the email notification options configured for the task. You can send email to different addresses based on the status of the task: - Error Email Notification. Sends email to the listed addresses when the task fails to complete. - Warning Email Notification. Sends email to the listed addresses when the task completes with errors. - Success Email Notification. Sends email to the listed addresses when the task completes without errors. Use commas to separate a list of email addresses. When you select this option, email notification options configured for the organization are not used.</td>
</tr>
</tbody>
</table>

5. Optionally, enter the following advanced options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing Commands</td>
<td>Command to run before the task.</td>
</tr>
<tr>
<td>Postprocessing Commands</td>
<td>Command to run after the task completes.</td>
</tr>
<tr>
<td>Update Columns</td>
<td>Database targets only. Temporary primary key columns to update target data.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Allow Null Updates</td>
<td>Salesforce targets only. Indicates if null values are allowed to replace existing values in the target. Select True to allow null values to replace existing values in the target. Default is False.</td>
</tr>
<tr>
<td>Create Success File</td>
<td>Salesforce targets only. Indicates if you want to create a success file for the task. Select True to create a success file. Default is False.</td>
</tr>
<tr>
<td>Target Batch Size</td>
<td>Salesforce targets only. The maximum number of records to include in each query that writes to the Salesforce target. Enter a number between 1 and 200. To process multiple upserts on a particular row in the same query, set the batch size to 1. Default is 200.</td>
</tr>
<tr>
<td>Assignment Rule Selection</td>
<td>Salesforce Case or Lead target objects only. Assignment rule to reassign attributes in records when inserting, updating, or upserting records: - None. Select to invoke no assignment rule. Default is None. - Default. Select to invoke the default assignment rule set for the organization. - Custom. Select to specify and invoke a custom assignment rule.</td>
</tr>
</tbody>
</table>

6. Click **Save**.

**RELATED TOPICS:**
- “Running or Scheduling a Data Synchronization Task” on page 115

**Promoting Data Loader Tasks to Data Synchronization Tasks**

You can promote data loader tasks to data synchronization tasks. After you promote the task, you cannot revert change.

**Note:** If you run a data loader task, and then promote the data loader task to a data synchronization task, the task appears in the activity log as a data loader task.

1. Click **Data Services > Data Synchronization > Promote DLS Tasks**.
2. On the Promote DLS Tasks page, move the data loader tasks that you want to promote to the Tasks to be Promoted area.
3. Click **OK**.
Viewing Data Synchronization Task Details

You can view details about a data synchronization task, including the source and target connections and the associated schedule. When you view the details, click Edit to modify the data synchronization task. You can also view the activity log for a data synchronization task.

- Click **Data Services > Data Synchronization**, and then click the data synchronization task name. The details appear on the View Data Synchronization Task page.

Copying a Data Synchronization Task

You can copy a data synchronization task. You might copy a task to create a different task with similar behavior. For example, you might create two data synchronization tasks to synchronize data in two different targets. Or, you might test changes on a copy of a task.

When you copy a task, the Data Synchronization service creates a new task name. The Data Synchronization service appends a number to the end of the task name. For example, if the original task name is DS1, the new task name is DS1_2.

**Note:** When you copy a task with a schedule, the schedule is removed from the copied task.

1. Click **Data Services > Data Synchronization**.
2. Click the **Make a Copy** icon next to the task name.

**Related Topics:**
- "Rules and Guidelines for Copying Objects" on page 156

Running a Data Synchronization Task

You can run a data synchronization task in the following ways:

- Manually
- On a schedule

**Rules and Guidelines for Running a Data Synchronization Task**

Use the following rules and guidelines when you run a data synchronization task:

- To run a data synchronization task that reads data from or writes data to a database, the Secure Agent and the corresponding database server must be running. The data synchronization task fails if either is not running.

- Verify that the source and target definitions are current. If the source or target no longer contains fields that are mapped in the field mapping, the data synchronization task fails.

- You cannot run multiple instances of a data synchronization task simultaneously. If you run a data synchronization task that is already running, the data synchronization task fails with the following error:

  Data loader task <data synchronization task name> failed to run. Another instance of the task is currently executing.
If you configured the data synchronization task to run on a schedule, increase the time interval between the scheduled tasks to prevent multiple instances of the data synchronization task from running simultaneously. If you run the data synchronization task manually, wait for the currently running instance of the data synchronization task to complete before starting it again. You can view currently running data synchronization tasks in the activity monitor.

- The data synchronization task does not load any data into an IBM DB2 target if one or more records fails to load.
- When you use an active plug-in with a data synchronization task that includes a custom source, the Data Synchronization service ignores the configured target option for the task and tries to insert data to the target.
- A data synchronization task that writes large amounts of data from Salesforce to an FTP/SFTP target file may fail unexpectedly with the following error:

  Error occurred while trying to close output file <output file name>

  This issue occurs when the FTP/SFTP server runs on a Windows 2000 32-bit machine. To resolve this issue, upgrade the server to a more recent version of Windows, and then run the task again.

Running a Data Synchronization Task Manually

Run a data synchronization task manually to immediately start the data synchronization task. You might want to run a data synchronization task manually for the following reasons:

- To verify that the data synchronization task is configured properly.
- To load data into the target occasionally. You may not need to update the target on regular intervals.

When you start a data synchronization task manually, the activity monitor appears and shows you details about the task.

1. Click Data Services > Data Synchronization.
2. On the Data Synchronization Tasks page, click the Run option for the data synchronization task that you want to run.

Running a Data Synchronization Task on a Schedule

Use a schedule to write data at a specified time or on regular intervals. Edit a data synchronization task to change the schedule associated with it.

1. To associate a schedule with a new data synchronization task, click Data Services > Data Synchronization > New, complete the preliminary wizard steps, and then click Schedule.
2. To associate a schedule with an existing data synchronization task, click Data Services > Data Synchronization, click the name of the data synchronization task, click Edit, and then click Schedule.
3. Select a schedule in the Schedule area, or click New to create a schedule.

Stopping a Data Synchronization Task

Click the Stop Job option in the activity monitor to stop a data synchronization task that is currently running.
Monitoring a Data Synchronization Task

You can view details about tasks that are currently running in the activity monitor. You can view details about completed tasks in the activity log.

Downloading Mapping XML

You can download the mapping XML for a data synchronization task. You might download mapping XML to import the mapping into PowerCenter.

1. Click **Data Services > Data Synchronization**.
2. Click **Download Mapping XML** next to the data synchronization task name.

Deleting a Data Synchronization Task

You can delete a data synchronization task at any time. If the data synchronization task is currently running, the task completes the task and cancels all scheduled tasks that have not started.

You cannot delete the task if it is included in a task flow. You must remove it from the task flow before you can delete the task. Before you delete a data synchronization task, verify that no users in the organization plan to use it. You cannot retrieve a data synchronization task after you delete it.

1. Click **Data Services > Data Synchronization**.
2. Click the **Delete** icon next to the data synchronization task name.
This chapter includes the following topics:

- PowerCenter Service Overview, 132
- PowerCenter Prerequisite Tasks, 132
- Supported Transformations and Other Mapping Objects, 133
- Pre- and Post-Session Commands, 134
- Rules and Guidelines for PowerCenter Sources and Targets, 134
- Configuring a PowerCenter Task, 135
- Viewing PowerCenter Task Details, 137
- Running a PowerCenter Task, 137
- Stopping a PowerCenter Task, 138
- Monitoring a PowerCenter Task, 138
- Deleting a PowerCenter Task, 138

PowerCenter Service Overview

The PowerCenter service allows you to import PowerCenter workflows into Informatica Cloud and run them as Informatica Cloud tasks. When you create a task, you can associate it with a schedule to run it at specified times or on regular intervals. Or, you can run it manually. You can monitor tasks that are currently running in the activity monitor and view logs about completed tasks in the activity log.

PowerCenter Prerequisite Tasks

Before you create a PowerCenter task, complete the following prerequisite tasks:

1. Verify that the sources, targets meet the requirements.
2. Download and run the Secure Agent if you read data from or write data to SAP, web services, databases, or flat files.
3. Validate the workflow in PowerCenter Workflow Manager before exporting the workflow to an XML file.
4. Verify that the XML file contains one workflow, one session, and one mapping.
5. Verify that all mapping objects in the workflow are supported in Informatica Cloud.
You can include the following transformations and other mapping objects in the workflow for a PowerCenter task:

- Flat file, database, SAP (SAPALEIDoc), web service, and XML sources and targets
- Aggregator transformation
- Expression transformation
- Filter transformation
- HTTP transformation
- Joiner transformation
- Lookup transformation
- Normalizer transformation
- Router transformation
- Salesforce Lookup transformation
- Salesforce Picklist transformation
- Salesforce Merge transformation
- SAP IDOC Interpreter transformation
- SAP IDOC Prepare transformation
- Sequence Generator transformation
- Sorter transformation
- Stored Procedure transformation
- Transaction Control transformation
- Union transformation
- Update Strategy transformation
- Web Services consumer transformation
- XML Parser transformation with file or database sources.
- XML Generator transformation with file or database sources.

If the workflow contains other mapping objects, the workflow import may fail.

**Exception Handling in Stored Procedures**

When you use a Stored Procedure transformation in a workflow for a PowerCenter task, the stored procedure must include exception handling. Exception handling can be as complex as necessary. Or, you can use the following simple example of exception handling code:

```sql
Exception when NO_DATA_FOUND then NULL;
END;
```

For example, you have the following stored procedure in a PowerCenter workflow:

```sql
CREATE OR REPLACE PROCEDURE SP_GETSAL_WITH_EXCEPTION (EMP_ID NUMBER, EMP_NAME OUT VARCHAR, SAL OUT NUMBER) AS BEGIN
    SELECT EMPNAME INTO EMP_NAME FROM EMPLOYEE WHERE EMPID=EMP_ID;
    SELECT SALARY INTO SAL FROM EMPLOYEE WHERE EMPID=EMP_ID;
END;
```
Before you export the workflow, add exception handling as follows:

```sql
CREATE OR REPLACE PROCEDURE SP_GETSAL_WITH_EXCEPTION (EMP_ID NUMBER, EMP_NAME OUT VARCHAR, SAL OUT NUMBER)
AS
BEGIN
    SELECT EMPNAME INTO EMP_NAME FROM EMPLOYEE WHERE EMPID=EMP_ID;
    SELECT SALARY INTO SAL FROM EMPLOYEE WHERE EMPID=EMP_ID;
    Exception
    when NO_DATA_FOUND
    then NULL;
END;
```

### Pre- and Post-Session Commands

You can include pre- and post-session commands in the session for a PowerCenter task. You can use pre- and post-session SQL or shell commands. You might use a pre- or post-session command to start FTP/SFTP scripts or stored procedures, rename or archive files, or run post-processing commands. Configure pre- and post-session commands in the PowerCenter session.

When you configure a pre- or post-session command, you can enter a single command, or you can call a batch file that contains a set of commands. To use a batch file, use the following guidelines:

- When you configure the pre- or post-session command, enter the complete path or directory along with the file name, such as c:/IC PowerCenter Task Commands/pre-session1.bat.
- In the batch file, use complete paths or directories.
- For a Secure Agent running on Windows, configure a network login for the Windows Secure Agent service. For more information, see “Configuring a Login for a Windows Secure Agent Service” on page 36.

### Rules and Guidelines for PowerCenter Sources and Targets

Use the following rules and guidelines for sources and targets in a PowerCenter task:

- Field names must contain only alphanumeric or underscore characters. Spaces are not allowed.
- Field names cannot start with a number.
- Each field name must be unique within each source and target object.
- The service truncates the data if the scale or precision of a numeric target column is less than the scale or precision of the corresponding source column.
- The import fails if Informatica Cloud does not support the source and target connection types that are imported with the PowerCenter workflow.
- Do not include NVARCHAR2 columns in Oracle targets. Due to an ODBC driver limitation, the PowerCenter Service truncates the last half of NVARCHAR2 data before writing it to Oracle targets.
Configuring a PowerCenter Task

You can create a PowerCenter task. To create a PowerCenter task, export the workflow from PowerCenter Repository Manager, and then import the PowerCenter workflow XML file in to Informatica Cloud using the New PowerCenter Task Wizard. After you import the file, you must map the imported connections to Informatica Cloud source, target, and, if applicable, lookup connections.

When you edit a PowerCenter task, you can change the name, description, schedule, and Secure Agent for a PowerCenter task. You can also update the task based on a new workflow XML file. When you upload a new workflow XML file to an existing PowerCenter task, the PowerCenter service deletes the old XML file and updates the PowerCenter task definition based on new XML content.

Rules and Guidelines for Configuring a PowerCenter Task

Use the following rules and guidelines when you configure a PowerCenter task:

- The PowerCenter workflow can contain only supported transformations and other mapping objects. For more information, see “Supported Transformations and Other Mapping Objects” on page 133.
- The PowerCenter workflow can contain a session task. Other tasks, such as Command and Decision tasks, are not supported.
- The workflow contain up to 10 partitions for sources and targets.
- The XML file can contain only one workflow, one session, and one mapping.
- The workflow import fails in the following cases:
  - The workflow XML file is empty.
  - The workflow XML file is not valid.
  - The workflow XML does not contain at least one source and target.
  - The workflow connections are not mapped to Informatica Cloud connections.
  - The workflow contains mapping objects that are not supported.
- When a PowerCenter XML file contains Web Service connection information, you can configure a Web Service connection in the PowerCenter task. If you configure a different connection type, the PowerCenter service uses Web Service connection information saved in the workflow.

Steps to Configure a PowerCenter Task

1. To create a PowerCenter task in Informatica Cloud, click Data Services > PowerCenter Service, and then click New.
   
   To edit a PowerCenter task, click Data Services > PowerCenter Service, click the name of the PowerCenter task, and then click Edit.

2. In the Task Details area, configure the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Name</td>
<td>Name of the PowerCenter task. The names of PowerCenter tasks must be unique within the organization. PowerCenter task names can contain alphanumeric characters, spaces, and the following special characters: _ , + -</td>
</tr>
</tbody>
</table>
### Field | Description
--- | ---
PowerCenter task names are not case sensitive. | 
Description | Description of the PowerCenter task. Maximum length is 255 characters.
Schedule | Determines how the task runs:
- To run the task manually, click Do Not Run This Task on a Schedule.
- To associate the task with a schedule, click Run This Task on Schedule, and select a schedule.
To create a schedule, click New.
Agent | The Secure Agent to run the task.
Workflow XML File | The PowerCenter workflow XML file associated with the task. Only the first 30 characters of the XML file name appear.
To upload a file, click Upload XML File. After you upload the workflow XML file, the connections and transformations appear in the Workflow XML File Details area.
To download the workflow XML file from Informatica Cloud, click Download XML File. You might download a file to import the workflow to the PowerCenter Designer for review.

3. Configure email notification options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Default Email Notification Options for my Organization</td>
<td>Use the email notification options configured for the organization.</td>
</tr>
</tbody>
</table>
| Use Custom Email Notification Options for this Task | Use the email notification options configured for the task. You can send email to different addresses based on the status of the task:
- Error Email Notification. Sends email to the listed addresses when the task fails to complete.
- Warning Email Notification. Sends email to the listed addresses when the task completes with errors.
- Success Email Notification. Sends email to the listed addresses when the task completes without errors.
Use commas to separate a list of email addresses. When you select this option, email notification options configured for the organization are not used. |

4. In the Connections area, map the connection references defined in the XML file to the connections in Informatica Cloud.
The connections references refer to source, target, and lookup connections defined in the workflow XML file. Select an existing connection from the drop-down list or click New Connection to create a new connection. To edit the selected connection, click Edit Connection.
The Transformations area displays all transformations defined in the workflow XML file.

5. Click Save.
Viewing PowerCenter Task Details

You can view details about a PowerCenter task, including the source and target connections and the associated schedule. When you view the details, click Edit to modify the PowerCenter task. You can also view the activity log for a PowerCenter task.

- Click Data Services > PowerCenter Service, and then click the PowerCenter task name.

Running a PowerCenter Task

You can run a PowerCenter task in the following ways:

- Manually
- On a schedule

Rules and Guidelines for Running a PowerCenter Task

Use the following rules and guidelines when you run a PowerCenter task:

- To run a PowerCenter task that reads data from or writes data to a database, the Secure Agent and the corresponding database server must be running. The PowerCenter task fails if either is not running.
- Ensure that the source and target definitions are current. If the source or target no longer contains fields that are mapped in the field mapping, the PowerCenter task fails.
- You cannot run multiple instances of a PowerCenter task simultaneously. If you run a PowerCenter task that is already running, the PowerCenter task fails.
- If the PowerCenter workflow uses the $PMSourceFileDir or $PMTargetFileDir variables to specify the source or target file directory location, you must copy the source or target files to the following directory:

  &lt;SecureAgent_InstallDirectory&gt;main\rdtmDir

  If you do not move the source or target files, the task fails.
- The PowerCenter task fails when the underlying PowerCenter workflow the following characteristics:
  - The PowerCenter session did not have the Enable High Precision session property enabled before it was exported.
  - The PowerCenter mapping contains an IIF expression with values of different datatypes, such as the following IIF expressions:

    ```
    \{IIF(ANNUALREVENUE &gt; 0, NAME)\}
    \{IIF(emplid_offset = 'Y', LINE_NO + 1, LINE_NO)\}
    ```

    To successfully run the PowerCenter task, enable high precision for the session in PowerCenter, export the workflow again, and then reimport the workflow.
- PowerCenter tasks run by a Secure Agent on Windows 7 (64 bit) might not complete. To resolve the issue, configure a network login for the Secure Agent service.

Running a PowerCenter Task Manually

Run a PowerCenter task manually to immediately start the task. You might want to run a task manually for the following reasons:

- To verify that the task is configured properly.
To load data into the target occasionally. You may not need to update the target on regular intervals.

When you start a task manually, the activity monitor appears and shows you details about the task.

1. Click Data Services > PowerCenter Service.
2. On the PowerCenter Tasks page, click the Run icon for the task that you want to run.

**Running a PowerCenter Task on a Schedule**

Use a schedule to write data at a specified time or on regular intervals. Edit a task to change the schedule associated with it.

1. To associate a schedule with a new task, click Data Services > PowerCenter Service > New.
   To associate a schedule with an existing data loader task, click Data Services > PowerCenter Service, click the name of the task, click Edit.
2. Select a schedule in the Schedule area, or click New to create a schedule.

**Stopping a PowerCenter Task**

Click the Stop Job option in the activity monitor to stop a task that is currently running.

**Monitoring a PowerCenter Task**

You can view details about tasks that are currently running in the activity monitor. You can view details about completed tasks in the activity log.

**Deleting a PowerCenter Task**

You can delete a task at any time. If the task is currently running, the PowerCenter service completes the task and cancels all scheduled tasks that have not started.

Before you delete a task, verify that no users in the organization plan to use it. You cannot retrieve a task after you delete it.

1. Click Data Services > PowerCenter Service.
2. Click the Delete icon next to the task name.
Common Task Configuration

This chapter includes the following topics:

- Common Task Configuration Overview, 139
- IDs for Related Salesforce Objects, 139
- Field Expressions, 141
- Data Filters, 145
- Email Notification for Tasks, 149
- Custom Views, 150
- Rules and Guidelines for Copying Objects, 156

Common Task Configuration Overview

This section discusses configuration procedures that are common to multiple types of tasks.

IDs for Related Salesforce Objects

Informatica Cloud identifies records of a Salesforce object based on one of the following types of IDs:

- Salesforce ID
  Salesforce generates an ID for each new record in a Salesforce object.

- External ID
  You can create a custom external ID field in the Salesforce object to identify records in the object. You might create an external ID to use the ID generated from a third-party application to identify records in the Salesforce object. You can use one or more external IDs to uniquely identify records in each Salesforce object.

If you create a data loader or data synchronization task that writes to a Salesforce target, the source must provide either the Salesforce IDs or the external IDs for the records in the Salesforce target object and applicable related objects. A related object is an object that is related to another object based on a relationship defined in Salesforce. The Data Loader service or Data Synchronization service uses the Salesforce ID or external ID to update changes to related objects.

If the source in a task contains external IDs for Salesforce objects, you must specify the external IDs for all related objects when you create the Salesforce target for the task. If you do not specify the external ID, Informatica Cloud requires the Salesforce ID to identify records in each related object.
Example of Configuring External IDs for Related Objects

You want to create a data loader task that upserts data from a flat file source into the Salesforce Contact object. You use external IDs to uniquely identify records in the Contact and Account Salesforce objects. Account is a related object to the Contact object based on the Contact relationship defined in Salesforce.

The Contact target object contains the following records:

<table>
<thead>
<tr>
<th>External Contact ID</th>
<th>Contact Name</th>
<th>Contact Street Address</th>
<th>Account</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtINFA001</td>
<td>Myra Hurley</td>
<td>100 Cardinal Way</td>
<td>ABC Tiles</td>
</tr>
<tr>
<td>ExtINFA002</td>
<td>Michael Buffet</td>
<td>202 Blume Drive</td>
<td>Music For You</td>
</tr>
<tr>
<td>ExtINFA003</td>
<td>Charles Tang</td>
<td>5002 Segway Street</td>
<td>Surplus Electronics</td>
</tr>
</tbody>
</table>

The Account related object contains the following records:

<table>
<thead>
<tr>
<th>External Account ID</th>
<th>Account Name</th>
<th>Billing City</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtAcct11</td>
<td>ABC Tiles</td>
<td>Los Angeles</td>
<td>Customer - Direct</td>
</tr>
<tr>
<td>ExtAcct12</td>
<td>Music For You</td>
<td>New York</td>
<td>Customer - Channel</td>
</tr>
<tr>
<td>ExtAcct13</td>
<td>Surplus Electronics</td>
<td>Atlanta</td>
<td>Customer - Direct</td>
</tr>
<tr>
<td>ExtAcct14</td>
<td>Richard Herlick</td>
<td>New Orleans</td>
<td>Customer - Direct</td>
</tr>
</tbody>
</table>

The flat file source contains the following records:

<table>
<thead>
<tr>
<th>External Contact ID</th>
<th>External Account ID</th>
<th>Contact Name</th>
<th>Contact Street Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtINFA001</td>
<td>ExtAcct11</td>
<td>Pam Dunn</td>
<td>100 Cardinal Way</td>
</tr>
<tr>
<td>ExtINFA002</td>
<td>ExtAcct13</td>
<td>Michael Buffet</td>
<td>202 Blume Drive</td>
</tr>
<tr>
<td>ExtINFA003</td>
<td>ExtAcct13</td>
<td>Charles Tang</td>
<td>5002 Segway Street</td>
</tr>
<tr>
<td>ExtINFA004</td>
<td>ExtAcct14</td>
<td>Henry Williams</td>
<td>40 B Street</td>
</tr>
</tbody>
</table>

You create the following data loader task:

- In step 1 of the Data Loader Task Wizard, you specify Salesforce as the target type and upsert as the task operation.
- In step 2 of the Data Loader Task Wizard, you specify the Salesforce Contact object as the target. You also click Related Objects and select External Account ID as the external ID for the Account related object.
- In step 3 of the Data Loader Task Wizard, you specify the flat file as the source.
- In step 5 of the Data Loader Task Wizard, you map the following fields:

<table>
<thead>
<tr>
<th>Source Field</th>
<th>Target Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Contact ID</td>
<td>ExternalContactId__c*</td>
</tr>
<tr>
<td>External Account ID</td>
<td>AccountId</td>
</tr>
<tr>
<td>Contact Name</td>
<td>Name</td>
</tr>
<tr>
<td>Contact Street Address</td>
<td>MailingStreet</td>
</tr>
</tbody>
</table>

*This is the external ID custom field for Contact in Salesforce.*

When you run the data loader task, the Data Loader service matches the External Contact ID for the ExtINFA001, ExtINFA002, and ExtINFA003 records in the Contact object, but finds no match for the ExtINFA004 record.

The Data Loader service makes the following updates to the Contact object:

- Changes the contact name for the existing ExtINFA001 record.
- Changes the account for the existing ExtINFA002 record.
- Inserts the new ExtINFA004 record.
After the data loader task completes, the Contact object contains the following records:

<table>
<thead>
<tr>
<th>External Contact ID</th>
<th>Contact Name</th>
<th>Contact Street Address</th>
<th>Account</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtINFA001</td>
<td>Pam Dunn</td>
<td>100 Cardinal Way</td>
<td>ABC Tiles</td>
</tr>
<tr>
<td>ExtINFA002</td>
<td>Michael Buffet</td>
<td>202 Blume Drive</td>
<td>Surplus Electronics</td>
</tr>
<tr>
<td>ExtINFA003</td>
<td>Charles Tang</td>
<td>5002 Segway Street</td>
<td>Surplus Electronics</td>
</tr>
<tr>
<td>ExtINFA004</td>
<td>Henry Williams</td>
<td>40 B Street</td>
<td>Richard Herlick</td>
</tr>
</tbody>
</table>

**Note:** If the external ID field contains blank values, the task fails.

## Field Expressions

You can transform the source data before loading it into the target. When you configure field mappings, you can specify an expression for each field mapping. You can map multiple source fields to the same target field. For example, you map SourceFieldA and SourceFieldB to TargetFieldC.

Informatica Cloud might suggest operations when you map multiple source fields to a single target field. For example, if you map multiple text fields to a target text field, Informatica Cloud concatenates the source text fields by default. You can change the default expression.

Informatica Cloud provides a transformation language that includes SQL-like functions to transform source data. Use these functions to write expressions, which modify data or test whether data matches the conditions that you specify.

To create a field expression:

1. To configure a field expression in a new task, click **Data Services > <service type> > New**, complete the preliminary wizard steps, and then click **Field Mapping**.
2. To configure a field expression in an existing data loader task, click **Data Services > <service type>**, click the name of the task, click **Edit**, and then click **Field Mapping**.
3. Click **Add Expression**.
4. Enter the new field expression.
5. Click **Validate Mapping** to validate the field mappings.
6. Click **OK**.

### Rules and Guidelines for Validating Field Mappings

Use the following rules and guidelines when you validate an expression in a field mapping:

- When you validate mappings, Informatica Cloud performs the following validations:
  - Verifies that the source and target fields in the task exist in the source or target. If the field does not exist, an error appears.
  - Verifies that all column datatypes are string and all field expressions contain string operations when the source and target are flat files.
  - Verifies that the correct parameters are used for each function and that the function is valid.

- The expression validator does not perform case-sensitive checks on field names.
- The expression validator verifies that the datatype of a field in an expression matches the datatype expected by the containing function. However, the expression validator does not check for incompatible datatypes between the following sets of objects:
  - Source and target fields of tasks.
  - Source field in a lookup condition and the lookup field.
  - Output of an expression or lookup and the target field.

  The expression or lookup with these incompatible datatypes may validate successfully, but, at runtime, the task fails and an error appears.

- If you map a string source field to a number target field, the validation succeeds. The service tries to convert the string to a number using the atoi (ASCII to Integer) C function.

- The expression validator does not validate lookups.

- If you validate the field mapping after you import a data loader process or previously saved the task, the following error may appear:

  **Missing data type on one or more source fields. Please click Refresh Fields button to reload fields.**

  To resolve this issue, refresh the fields. When you refresh the fields, Informatica Cloud retrieves and updates the latest field attributes, including the field datatypes.

### Transformation Language Components for Expressions

The transformation language includes the following components to create simple or complex expressions:

- **Fields.** Use the name of a source field to refer to the value of the field.
- **Literals.** Use numeric or string literals to refer to specific values.
- **Functions.** Use these SQL-like functions to change data in a task.
- **Operators.** Use transformation operators to create expressions to perform mathematical computations, combine data, or compare data.
- **Constants.** Use the predefined constants to reference values that remain constant, such as TRUE.

### Expression Syntax

You can create a simple expression that only contains a field, such as ORDERS, or a numeric literal, such as 10. You can also write complex expressions that include functions nested within functions, or combine different fields using the transformation language operators.

**Note:** Although the transformation language is based on standard SQL, there are differences between the two languages.

### String and Numeric Literals

You can include numeric or string literals.

Enclose string literals within single quotation marks. For example:

```plaintext
'Alice Davis'
```

String literals are case sensitive and can contain any character except a single quotation mark. For example, the following string is not allowed:

```
'Joan's car'
```

To return a string containing a single quotation mark, use the CHR function:

```plaintext
'Joan' || CHR(39) || 's car'
```
Do not use single quotation marks with numeric literals. Just enter the number you want to include. For example:

```
.05
```
or
```
$$Sales_Tax
```

**Rules and Guidelines for Expressions**

Use the following rules and guidelines when you write expressions:
- For each source field, you can perform a lookup or create an expression. You cannot do both.
- You cannot use strings in numeric expressions.
  
  For example, the expression `1 + '1'` is not valid because you can only perform addition on numeric datatypes.
  You cannot add an integer and a string.
- You cannot use strings as numeric parameters.
  
  For example, the expression `SUBSTR(TEXT_VAL, '1', 10)` is not valid because the SUBSTR function requires an integer value, not a string, as the start position.
- You cannot mix datatypes when using comparison operators.
  
  For example, the expression `123.4 = '123.4'` is not valid because it compares a decimal value with a string.
- You can pass a value from a field, literal string or number, or the results of another expression.
- Separate each argument in a function with a comma.
- Except for literals, the transformation language is not case sensitive.
- The colon (:), comma (,), and period (.) have special meaning and should be used only to specify syntax.
- Informatica Cloud Services treat a dash (-) as a minus operator.
- If you pass a literal value to a function, enclose literal strings within single quotation marks. Do not use quotation marks for literal numbers. Informatica Cloud Services treat any string value enclosed in single quotation marks as a character string.
- Do not use quotation marks to designate fields.
- You can nest multiple functions within an expression. Informatica Cloud Services evaluate the expression starting with the innermost function.

**Adding Comments to Expressions**

You can use the following comment specifiers to insert comments in expressions:
- Two dashes:
  
  `-- These are comments`
- Two forward slashes:
  
  `// These are comments`

Informatica Cloud Services ignore all text on a line preceded by comment specifiers. For example, to concatenate two strings, enter the following expression with comments in the middle of the expression:

```
-- This expression concatenates first and last names for customers:
FIRST_NAME // First names from the CUST table
|| // Concat symbol
LAST_NAME // Last names from the CUST table
// Joe Smith Aug 18 1998
```

Informatica Cloud Services ignore the comments and evaluates the expression as follows:

```
FIRST_NAME || LAST_NAME
```
You cannot continue a comment to a new line:

-- This expression concatenates first and last names for customers:
FIRST_NAME -- First names from the CUST table
|| // Concat symbol
LAST_NAME // Last names from the CUST table

Joe Smith Aug 18 1998

In this case, Informatica Cloud Services do not validate the expression because the last line is not a valid expression.

Reserved Words

Some keywords, such as constants, operators, and system variables, are reserved for specific functions. These include:

- :EXT
- :INF
- :LKP
- :MCR
- :SD
- :SEQ
- :SP
- :TD
- AND
- DD_DELETE
- DD_INSERT
- DD_REJECT
- DD_UPDATE
- FALSE
- NOT
- NULL
- OR
- PROC_RESULT
- SESSSTARTTIME
- SPOUTPUT
- SYSDATE
- TRUE
- WORKFLOWSTARTTIME

The following words are reserved for Informatica Cloud:

- ABORTED
- DISABLED
- FAILED
- NOTSTARTED
- STARTED
- STOPPED
Data Filters

You can create the following type of data filters for any type of task:

- Simple
- Advanced

You can create a set of data filters for each object included in a data replication, data loader, or data synchronization task. Each set of data filters act independently of the other sets.

Simple Data Filters

Create one or more simple data filters. When you create multiple simple data filters, the associated service creates an AND operator between the filters and loads rows that apply to all simple data filters. For example, you load rows from the Account Salesforce object to a database table. However, you want to load only accounts that have greater than or equal to $100,000 in annual revenue and that have more than 500 employees. You configure the following simple data filters:

<table>
<thead>
<tr>
<th>Field</th>
<th>Operator</th>
<th>Field Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnnualRevenue</td>
<td>greater than or equals</td>
<td>100000</td>
</tr>
<tr>
<td>NumberOfEmployees</td>
<td>greater than</td>
<td>500</td>
</tr>
</tbody>
</table>

To configure a simple data filter:

1. **To configure a simple data filter in a new task**, click **Data Services** > **<service type>** > **New**, complete the preliminary wizard steps, and then click **Data Filters**.

   To configure a simple data filter in an existing task, click **Data Services** > **<service type>**, click the name of the task, click **Edit**, and then click **Data Filters**.

   **<service type>** is the name of any of the services, such as Data Loader service.

2. **On the Data Filters page**, click **Simple**, and then click **New** to create a data filter. The Data Filter dialog box appears.

3. **When you configure a data filter**, specify the object on which to create the data filter. You create separate data filters for each source object included in the task.

4. **Enter the filter condition based on the field, operator, and field value.**

5. **Click OK.**

6. **Repeat steps 3 to 5 to create another simple data filter.**

7. **To delete a data filter**, click the **Delete** icon next to the data filter.

8. **Click Next.**
Advanced Data Filters

Create an advanced data filter to create complex expressions that use AND, OR, or nested conditions. You enter one expression that contains all filters. The expression that you enter becomes the WHERE clause in the query used to retrieve records from the source.

For example, you load rows from the Account Salesforce object to a database table. However, you want to load records where the billing state is California or New York and the annual revenue is greater than or equal to $100,000. You configure the following advanced filter expression:

\[(\text{BillingState} = \text{CA} \text{ OR BillingState} = \text{NY}) \text{ AND (AnnualRevenue} \geq 100000\)]

When you create a data filter on a Salesforce object, the corresponding service generates a SOQL query with a WHERE clause. The WHERE clause represents the data filter. The SOQL query must be less than 5,000 characters. If the query exceeds the character limit, the following error appears:

Salesforce SOQL limit of 5000 characters has been exceeded for the object: <Salesforce object>. Please exclude more fields or decrease the filters.

To configure an advanced data filter:

1. To configure an advanced data filter in a new task, click Data Services > <service type> > New, complete the preliminary wizard steps, and then click Data Filters.
2. To configure an advanced data filter in an existing task, click Data Services > <service type>, click the name of the task, click Edit, and then click Data Filters.
3. When you configure a data filter, specify the object on which to create the data filter.
   You create separate data filters for each source object included in the task.
4. Enter the filter expression.
   Click the field name to add the field to the expression.
5. Click OK.
6. To delete a data filter, click the Delete icon next to the data filter.
7. Click Next.

Data Filter Operators

You can use the specified operators for each of the following general field types:

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Supported Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>Equals, Not Equals, Is Null, Is Not Null</td>
</tr>
<tr>
<td>Currency</td>
<td>Equals, Not Equals, Less Than, Less Than or Equals, Greater Than, Greater Than or Equals, Is Null, Is Not Null</td>
</tr>
<tr>
<td>Date</td>
<td>Equals, Not Equals, Less Than, Less Than or Equals, Greater Than, Greater Than or Equals, Is Null, Is Not Null</td>
</tr>
<tr>
<td>Datetime</td>
<td>Equals, Not Equals, Less Than, Less Than or Equals, Greater Than, Greater Than or Equals, Is Null, Is Not Null</td>
</tr>
</tbody>
</table>
### Field Types and Supported Operators

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Supported Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double</td>
<td>Equals, Not Equals, Less Than, Less Than or Equals, Greater Than, Greater Than or Equals, Is Null, Is Not Null</td>
</tr>
<tr>
<td>ID</td>
<td>Equals, Not Equals, Is Null, Is Not Null</td>
</tr>
<tr>
<td>Int</td>
<td>Equals, Not Equals, Less Than, Less Than or Equals, Greater Than, Greater Than or Equals, Is Null, Is Not Null</td>
</tr>
<tr>
<td>Reference</td>
<td>Equals, Not Equals, Is Null, Is Not Null</td>
</tr>
<tr>
<td>String</td>
<td>Equals, Not Equals, Starts With, Ends With, Contains, Is Null, Is Not Null</td>
</tr>
<tr>
<td>Textarea</td>
<td>Equals, Not Equals, Starts With, Ends With, Contains, Is Null, Is Not Null</td>
</tr>
<tr>
<td>All other field types</td>
<td>Equals, Not Equals, Is Null, Is Not Null</td>
</tr>
</tbody>
</table>

### Data Filter Variables

You can use the following variables in simple and advanced data filter conditions:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LastRunDate</td>
<td>The last date in GMT time zone when the task ran successfully. Does not include time. For example, 2008-09-24. Can be used as a value for filter where the field type is 'DATE.'</td>
</tr>
<tr>
<td>$LastRunTime</td>
<td>The last date and time in GMT time zone when the task ran successfully. For example, 2008-09-24 15:23:23. Can be used as a value for filter where the field type is 'DATETIME.' You cannot use the $LastRunTime variable with DATE fields.</td>
</tr>
</tbody>
</table>

For example, you can include the following simple filter condition:

```
LastModifiedDate > $LastRunTime
```

**Note:** Consider the time zone differences when comparing dates across time zones. The date and time of the $LastRunDate and $LastRunTime variables are based on the time zone set in the Informatica Cloud application. The date and time of the actual job is based on the GMT time zone for Salesforce sources and the database server for database sources. The difference in the time zones may yield unexpected results.

### Rules and Guidelines for Data Filters

Use the following rules and guidelines when you configure a simple or advanced data filter for each applicable task type:

<table>
<thead>
<tr>
<th>Data Filter Types</th>
<th>Task Types</th>
<th>Rule / Guideline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>Data synchronization</td>
<td>You cannot create a simple data filter in a data synchronization task that includes a flat file source. You can create an advanced data filter.</td>
</tr>
<tr>
<td>Simple</td>
<td>Data synchronization</td>
<td>If you change the datatype of a Microsoft Access or ODBC source column to binary, varbinary, longvarbinary, or ntext in a data synchronization task, you cannot create a data filter on the column.</td>
</tr>
<tr>
<td>Data Filter Types</td>
<td>Task Types</td>
<td>Rule / Guideline</td>
</tr>
<tr>
<td>------------------</td>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Simple</td>
<td>All</td>
<td>The list of available operators depends on the datatype of the field included in the data filter. Some operators do not apply to all fields included in data filters.</td>
</tr>
<tr>
<td>Advanced</td>
<td>All</td>
<td>The service does not apply filters with Equals, Starts With, or Ends With operators and string fields that contain data that starts or ends with a single quotation mark. To filter these records, use the Contains operator.</td>
</tr>
<tr>
<td>Simple</td>
<td>Advanced</td>
<td>When you convert a simple data filter to an advanced data filter, you cannot convert the advanced data filter back to a simple data filter.</td>
</tr>
<tr>
<td>Simple</td>
<td>All</td>
<td>If you create a simple filter on a date field in an Oracle source, the date field must use the default Oracle format DD-MON-YY. To create a filter on a date field with a non-default date format, use an advanced data filter. For example, an Oracle date field that has the date format 'yyyy/mm/dd:hh:mi:ssam.' You create the following advanced filter: &quot;DATE_FIELD&quot; &gt; TO_DATE('1998/05/31:12:00:00AM', 'yyyy/mm/dd:hh:mi:ssam').</td>
</tr>
<tr>
<td>Advanced</td>
<td>All</td>
<td>You can select Equals, Not Equals, Is Null, or Is Not Null operators on fields of the Other datatype.</td>
</tr>
<tr>
<td>Simple</td>
<td>All</td>
<td>The data filter must contain valid SQL or SOQL operators.</td>
</tr>
</tbody>
</table>

When you write Salesforce data to a database target, verify that the Salesforce data uses the following required formats for date and time datatypes: Date (yyyy-MM-dd) and DateTime (yyyy-MM-dd HH:mm:ss). If a record contains the date and time in a different format, the service rejects the row. If the Salesforce source contains a date and no time for the datetime datatype, the service appends '00:00:00' at the end of the date value to ensure the value is in the required format. When you write to a database target, the service converts the Salesforce date and datetime data to the correct format expected by the database.

When you run a task that contains a filter on a date field from an Microsoft Access, MySQL, or Oracle database and the connection type is an ODBC connection, the task fails with an error similar to one of the following errors:

```
RR_4035 SQL Error [ FnName: Execute -- [Microsoft][ODBC Microsoft Access Driver] Data type mismatch in criteria expression.].

RR_4035 SQL Error [ FnName: Execute -- [Oracle][ODBC][Ora]ORA-01843: not a valid month ].
```

By default, the filter applies a double quotation mark to escape the column name in an ODBC connection, which causes the error. To resolve the error, create an advanced data filter and apply the correct escape character to the column names.

When you run a task that contains a filter on a date field from an Microsoft Access, MySQL, or Oracle database and the connection type is an ODBC connection, the task fails with an error similar to one of the following errors:

```
RR_4035 SQL Error [ FnName: Execute -- [Microsoft][ODBC Microsoft Access Driver] Data type mismatch in criteria expression.].
RR_4035 SQL Error [ FnName: Execute -- [Oracle][ODBC][Ora]ORA-01843: not a valid month ].
```

By default, the filter applies a double quotation mark to escape the column name in an ODBC connection, which causes the error. To resolve the error, create an advanced data filter and apply the correct escape character to the column names.

Salesforce fields of LongTextArea datatype do not show up in the list of fields of a filter.

When you enter more than one simple data filter, the service filters rows that meet the requirements of all data filters.

When you convert a simple data filter to an advanced data filter, you cannot convert the advanced data filter back to a simple data filter.

When you run a task that contains a filter on a date field from an Microsoft Access, MySQL, or Oracle database and the connection type is an ODBC connection, the task fails with an error similar to one of the following errors:

```
RR_4035 SQL Error [ FnName: Execute -- [Microsoft][ODBC Microsoft Access Driver] Data type mismatch in criteria expression.].
RR_4035 SQL Error [ FnName: Execute -- [Oracle][ODBC][Ora]ORA-01843: not a valid month ].
```

By default, the filter applies a double quotation mark to escape the column name in an ODBC connection, which causes the error. To resolve the error, create an advanced data filter and apply the correct escape character to the column names.

If you create a simple filter on a date field in an Oracle source, the date field must use the default Oracle format DD-MON-YY. To create a filter on a date field with a non-default date format, use an advanced data filter. For example, an Oracle date field that has the date format 'yyyy/mm/dd:hh:mi:ssam.' You create the following advanced filter:

```
"DATE_FIELD" > TO_DATE('1998/05/31:12:00:00AM', 'yyyy/mm/dd:hh:mi:ssam').
```
<table>
<thead>
<tr>
<th>Data Filter Types</th>
<th>Task Types</th>
<th>Rule / Guideline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced</td>
<td>All</td>
<td>You cannot include simple and advanced data filters in the same task.</td>
</tr>
<tr>
<td>Simple Advanced</td>
<td>All</td>
<td>A task fails if the fields included in the data filter no longer exist or if the datatypes of the fields change. If a datatype changes, edit the task.</td>
</tr>
<tr>
<td>Simple Advanced</td>
<td>All</td>
<td>When you include a Salesforce field of URL datatype, exclude “http://” from the value. For example, if the value is <a href="http://www.informatica.com">http://www.informatica.com</a>, enter <a href="http://www.informatica.com">www.informatica.com</a>.</td>
</tr>
<tr>
<td>Simple Advanced</td>
<td>All</td>
<td>When you include a Salesforce field with the Phone datatype, enter a value with the following syntax (XXX) XXX-XXXX. For example, enter (555) 555-1212. If you provide an incorrect syntax, the service ignores the filter.</td>
</tr>
<tr>
<td>Simple Advanced</td>
<td>All</td>
<td>When you include a Salesforce ID field in a filter, enter the exact ID value. If you enter a dummy ID value, the SOQL query fails.</td>
</tr>
<tr>
<td>Simple Advanced</td>
<td>All</td>
<td>You can only use IS NULL and LIKE operators in data filters for fields of the Text, Ntext, and Image datatypes.</td>
</tr>
<tr>
<td>Simple Advanced</td>
<td>All</td>
<td>You cannot include UTF-8 characters in a data filter on a Microsoft SQL Server or an Oracle source.</td>
</tr>
<tr>
<td>Simple Advanced</td>
<td>All</td>
<td>If you specify a date and no time for a date/time filter, the service uses 00:00:00 (12:00:00 a.m.) as the time.</td>
</tr>
</tbody>
</table>

### Email Notification for Tasks

You can configure email notification for a task or task flow. When you configure email notification for the task, Informatica Cloud uses the email notification options configured for the task instead of the email notification options configured for the organization.

You can configure email notification all task types except Data Loader.

1. To configure email notification in a new task, click **Data Services > <service type> > New**, complete the preliminary wizard steps, and then click **Schedule**.
   
   To configure email notification in an existing task, click **Data Services > <service type>**, click the name of the task, click **Edit**, and then click **Schedule**.

   `<service type>` is the name of any of the services, except Data Loader service.
2. Configure email notification options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Default Email Notification Options for my Organization</td>
<td>Use the email notification options configured for the organization.</td>
</tr>
</tbody>
</table>
| Use Custom Email Notification Options for this Task | Use the email notification options configured for the task. You can send email to different addresses based on the status of the task:  
- Error Email Notification. Sends email to the listed addresses when the task fails to complete.  
- Warning Email Notification. Sends email to the listed addresses when the task completes with errors.  
- Success Email Notification. Sends email to the listed addresses when the task completes without errors.  
Use commas to separate a list of email addresses.  
When you select this option, email notification options configured for the organization are not used. |

3. Click **Save**.

## Custom Views

You can configure custom views for connections, tasks, and task flows. By default, Informatica Cloud displays all objects of the selected type in the organization. A custom view lists a subset of the objects based on the configured options. Use custom views to help manage connections, tasks, and task flows.

You can create a private view. A private view is a view that only you can use, edit or delete. You can also create a public view. Anyone in your organization can use, edit, or delete a public view.

You can use custom views for the following objects:

- Connections  
- Data assessment tasks  
- Data loader tasks  
- Data replication tasks  
- Data synchronization tasks  
- PowerCenter tasks  
- Task flows

### Configuring a Custom View

You can configure custom views for connections, tasks, and task flows to display a subset of all connections, tasks, or task flows in the organization.

When you create a custom view, you configure the following information:

- **Requirements.** Details used to limit the objects that the view displays, such as Created By or Name Contains.  
- **Display details.** Details that the view displays, such as Connection Type or Last Run.  
- **Sorting.** Sort by column and sort order used to order the objects that the view displays.
When you configure more than one requirement for a view, Informatica Cloud displays the objects that fulfill all requirements. For example, you configure a view to display data synchronization tasks that use a Salesforce source connection type and a Salesforce target connection type. The view displays data synchronization tasks that use both Salesforce source and target connections. The view does not display tasks that use a Salesforce source connection with other target connection types.

To configure a custom view:

1. To configure a view for connections, click **Configuration > Connections**.
   To configure a view for tasks, click **Data Services > <service type>**.
   To configure a view for task flows, click **Configuration > Task Flows**.
2. On the Connection, Task, or Task Flow page, click **New View** to create a new view.
   To edit an existing view, select the view, then click **Edit View**.
3. On the New View page, configure the following View Details:

<table>
<thead>
<tr>
<th>View Detail</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>View Name</strong></td>
<td>Required. Name of the view. The name must be unique for the service. For example, you can create only one data synchronization task view named Salesforce, but you can create a view named Salesforce for connections and other services. The name can contain alphanumeric characters, spaces, and the following special characters: _ . + - Names are not case sensitive.</td>
</tr>
<tr>
<td><strong>View Type</strong></td>
<td>Indicates the type of the view you are creating, such as a Connections View or Data Synchronization Tasks view.</td>
</tr>
</tbody>
</table>
   | **View Visibility** | Determines the visibility of the view:  
   - Public. Anyone in the organization can use, edit, or delete the view.   
   - Private. Only you can use, edit, or delete the view.                      |

4. If you are configuring a view for connections, select the Connection Details that you want to use.
   If you are configuring a view for tasks, select the Task, Source, or Target Details you want to use.
   If you are configuring a view for task flows, select the Task Flow Details that you want to use.

   For a connection view, you can use any of the following Connection Details:

<table>
<thead>
<tr>
<th>Connection Details</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name contains</strong></td>
<td>Partial name of the connection. Enter the characters you want to use. This field is case sensitive.</td>
</tr>
<tr>
<td><strong>Description contains</strong></td>
<td>Partial description of the connection. Enter the characters you want to use. This field is case sensitive.</td>
</tr>
<tr>
<td><strong>Secure Agent</strong></td>
<td>Secure Agent used in the task. If you do not want to specify a Secure Agent, select Any Secure Agent. PowerCenter task views only.</td>
</tr>
<tr>
<td><strong>Created by</strong></td>
<td>The user who created the connection. Select a user name. If you do not want to specify a user, select Any User In My Organization.</td>
</tr>
<tr>
<td><strong>Last updated by</strong></td>
<td>The user who last updated the task. Select a user name.</td>
</tr>
</tbody>
</table>
### Connection Details

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>If you do not want to specify a user, select Any User In My Organization.</td>
<td></td>
</tr>
<tr>
<td>Created between Date range when the connection was created.</td>
<td></td>
</tr>
<tr>
<td>Updated between Date range when the connection was updated.</td>
<td></td>
</tr>
<tr>
<td>Connection Type Connection type. Select a connection type.</td>
<td></td>
</tr>
<tr>
<td>If you do not want to specify a connection type, select Any Connection Type.</td>
<td></td>
</tr>
<tr>
<td>When you select a connection type, you can use any related connection properties that display.</td>
<td></td>
</tr>
</tbody>
</table>

For a task view, you can use any of the following Task Details:

<table>
<thead>
<tr>
<th>Task Details</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name contains</td>
<td>Partial name of the task. Enter the characters you want to use. This field is case sensitive.</td>
</tr>
<tr>
<td>Description contains</td>
<td>Partial description of the task. Enter the characters you want to use. This field is case sensitive.</td>
</tr>
<tr>
<td>Source connection</td>
<td>Source connection used in the task. If you do not want to specify a source connection, select Include Any Connection. Data assessment task views only.</td>
</tr>
<tr>
<td>Source object</td>
<td>Source object used in the task. If you do not want to specify a source object, select Any Object. Data assessment task views only.</td>
</tr>
<tr>
<td>Target prefix contains</td>
<td>Partial target prefix. Enter the characters you want to use. Data replication task views only.</td>
</tr>
<tr>
<td>Created by</td>
<td>The user who created the task. Select a user name. If you do not want to specify a user, select Any User In My Organization.</td>
</tr>
<tr>
<td>Last updated by</td>
<td>The user who last updated the task. Select a user name. If you do not want to specify a user, select Any User In My Organization.</td>
</tr>
<tr>
<td>Created between</td>
<td>Date range when the task was created.</td>
</tr>
<tr>
<td>Updated between</td>
<td>Date range when the task was updated.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Schedule associated with the task. Select a schedule. If you do not want to specify a schedule, select Do Not Filter on Schedule.</td>
</tr>
</tbody>
</table>
For a task view, you can also use any of the following Source Details:

<table>
<thead>
<tr>
<th>Source Details</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any Connection and Type</td>
<td>Includes all source connections in the organization. Select if you do not want to specify a connection or connection type.</td>
</tr>
<tr>
<td>Filter by Source Connection Type</td>
<td>Filters by the source connection type used in the task. Select to display tasks that use the selected source connection type.</td>
</tr>
<tr>
<td>Filter by Source Connection</td>
<td>Filters by the source connection used in the task. Select to display tasks that use the selected source connection.</td>
</tr>
</tbody>
</table>

For a task view, you can also use any of the following Target Details:

<table>
<thead>
<tr>
<th>Target Details</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any Connection and Type</td>
<td>Includes all target connections in the organization. Select if you do not want to specify a connection or connection type.</td>
</tr>
<tr>
<td>Filter by Target Connection Type</td>
<td>Filters by the target connection type used in the task. Select to display tasks that use the selected target connection type.</td>
</tr>
<tr>
<td>Filter by Target Connection</td>
<td>Filters by the target connection used in the task. Select to display tasks that use the selected target connection.</td>
</tr>
</tbody>
</table>

For a task flow view, you can use any of the following Task Flow Details:

<table>
<thead>
<tr>
<th>Task Flow Details</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name contains</td>
<td>Partial name of the task flow. Enter the characters you want to use. This field is case sensitive.</td>
</tr>
<tr>
<td>Description contains</td>
<td>Partial description of the task flow. Enter the characters you want to use. This field is case sensitive.</td>
</tr>
<tr>
<td>Created by</td>
<td>The user who created the task flow. Select a user name. If you do not want to specify a user, select Any User In My Organization.</td>
</tr>
<tr>
<td>Last updated by</td>
<td>The user who last updated the task flow. Select a user name. If you do not want to specify a user, select Any User In My Organization.</td>
</tr>
<tr>
<td>Created between</td>
<td>Date range when the task flow was created.</td>
</tr>
<tr>
<td>Updated between</td>
<td>Date range when the task flow was updated.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Schedule associated with the task flow. Select a schedule. If you do not want to specify a schedule, select Do Not Filter on Schedule.</td>
</tr>
</tbody>
</table>

5. To configure the fields that the view displays, select the fields you want to use in Available Fields and copy the fields to Selected Fields.

**Note:** The view always displays the name of the connection or task.
For a connection view, you can use any of the following connection display fields:

<table>
<thead>
<tr>
<th>Available Fields for Connections</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Description of the connection.</td>
</tr>
<tr>
<td>Created Date</td>
<td>Date the connection was created.</td>
</tr>
<tr>
<td>Updated Date</td>
<td>Date the connection was last updated.</td>
</tr>
<tr>
<td>Created By</td>
<td>User who created the connection.</td>
</tr>
<tr>
<td>Updated By</td>
<td>User who last updated the connection.</td>
</tr>
<tr>
<td>Type</td>
<td>Connection type, such as Flat File or Oracle.</td>
</tr>
<tr>
<td>User Name</td>
<td>User name used in the connection.</td>
</tr>
<tr>
<td>Host Name</td>
<td>Host name used in the connection.</td>
</tr>
<tr>
<td>Port</td>
<td>Port number used in the connection.</td>
</tr>
<tr>
<td>Instance Name</td>
<td>Instance name used in the connection.</td>
</tr>
<tr>
<td>Database</td>
<td>Database name used in the connection.</td>
</tr>
<tr>
<td>Schema</td>
<td>Schema name used in the connection.</td>
</tr>
<tr>
<td>Service URL</td>
<td>Service URL used in the connection.</td>
</tr>
<tr>
<td>Secure Agent</td>
<td>Secure Agent used in the connection.</td>
</tr>
</tbody>
</table>

For a task view, you can use any of the following task display fields. Available fields might differ based on the type of task view you are configuring.

<table>
<thead>
<tr>
<th>Available Fields for Tasks</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Description of the task.</td>
</tr>
<tr>
<td>Created Date</td>
<td>Date the task was created.</td>
</tr>
<tr>
<td>Updated Date</td>
<td>Date the task was last updated.</td>
</tr>
<tr>
<td>Created By</td>
<td>User who created the task.</td>
</tr>
<tr>
<td>Updated By</td>
<td>User who last updated the task.</td>
</tr>
<tr>
<td>Source Connection</td>
<td>Source connection name.</td>
</tr>
<tr>
<td>Source Object</td>
<td>Source object used in a data assessment task.</td>
</tr>
<tr>
<td>Target Connection</td>
<td>Target connection name.</td>
</tr>
<tr>
<td>Available Fields for Tasks</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Objects</td>
<td>Objects used in the data replication task.</td>
</tr>
<tr>
<td>Operation</td>
<td>Task operation used in the data synchronization or data loader task.</td>
</tr>
<tr>
<td>Source Connection Type</td>
<td>Source connection type used in the task.</td>
</tr>
<tr>
<td>Target Connection Type</td>
<td>Target connection type used in the task.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Schedule used in the task.</td>
</tr>
<tr>
<td>Target Prefix</td>
<td>Target prefix used in the data replication task.</td>
</tr>
<tr>
<td>Replicate All Objects</td>
<td>Indicates if all objects are replicated in the data replication task.</td>
</tr>
<tr>
<td>Secure Agent</td>
<td>Secure Agent used in the PowerCenter task.</td>
</tr>
<tr>
<td>PowerCenter XML</td>
<td>Name of the PowerCenter XML file used in the PowerCenter task.</td>
</tr>
<tr>
<td>Last Run</td>
<td>Last time the task ran.</td>
</tr>
</tbody>
</table>

For a task flow view, you can use any of the following task flow display fields:

<table>
<thead>
<tr>
<th>Available Fields for Task Flows</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Description of the task flow.</td>
</tr>
<tr>
<td>Created Date</td>
<td>Date the task flow was created.</td>
</tr>
<tr>
<td>Updated Date</td>
<td>Date the task flow was last updated.</td>
</tr>
<tr>
<td>Created By</td>
<td>User who created the task flow.</td>
</tr>
<tr>
<td>Updated By</td>
<td>User who last updated the task flow.</td>
</tr>
<tr>
<td>Tasks</td>
<td>Tasks in the task flow.</td>
</tr>
<tr>
<td>Schedule</td>
<td>Schedule used in the task flow.</td>
</tr>
<tr>
<td>Last Run</td>
<td>Last time the task flow ran.</td>
</tr>
</tbody>
</table>

6. To change the column display order, use the navigation arrows.
7. Select a **Sort by Column**.
   You can use any column selected to display.
8. Select a **Sort Order**.
   You can sort the results in ascending or descending alphabetic order.
9. Click **OK**.
   Informatica Cloud saves and displays the view.
Working with a Custom View

You can use private views that you create. You can also use public views that are shared within the organization. After you select a view, you can click a column name to temporarily change the sort order of the view. Informatica Cloud displays the view with the original sort order if you leave and return to the page or if you switch to a different view and return to the view.

When you select a view, it becomes part of your user preferences. You can edit your user preferences to define your default views.

To select a custom view:

1. To select a custom view for connections, click Configuration > Connections. To select a custom view for tasks, click Data Services > <service type>. To select a custom view for task flows, click Configuration > Task Flows.

2. From the View menu, select the view you want to use.

The view displays. To see how the view is configured, or to edit the view, click Edit View. You can click on a column name to sort the view. An arrow appears to indicate the sort order.

Related Topics:
- “Editing User Preferences” on page 6

Deleting a Custom View

You can delete a custom view. You can delete a private view if you created it. You can also delete a public view. Before you delete a public view, make sure no one in the organization is using it. You cannot retrieve a view after it has been deleted.

To delete a custom view:

1. To delete a view for connections, click Configuration > Connections. To delete a view for tasks, click Data Services > <service type>. To delete a view for task flows, click Configuration > Task Flows.

2. Select the view, then click Edit View.

3. On the Edit View page, click Delete, and then click OK.

Rules and Guidelines for Copying Objects

Use the following rules and guidelines when copying objects:

- You can copy the following types of objects:
  - Connections
  - Custom sources
  - Data assessment tasks
  - Data replication tasks
  - Data synchronization tasks

- When naming the copied object, Informatica Cloud uses the same name as the original object, but appends an underscore and a number to the end of the original name. If you make multiple copies of the same object, Informatica Cloud increments the appended number by 1.
• When you copy a task that is associated with a schedule, the copied task is not associated with the schedule.
• When you copy an object, the copied object has the same permissions as the original object.
Monitoring Jobs

This chapter includes the following topics:

- Monitoring Jobs Overview, 158
- Activity Log, 158
- Activity Monitor, 162
- Dashboard, 162
- Data Files, 163
- Error Log Files, 163

Monitoring Jobs Overview

A job is an instance of a task. You can use the following tools to monitor jobs:

- **Activity log.** Provides run-time details about jobs that have completed.
- **Activity monitor.** Provides run-time details about jobs that are running.
- **Dashboard.** Provides charts that summarize the results of data assessment jobs.
- **Data files.** Displays data assessment data files available for download.
- **Error log files.** Contains information about rows that were not written to the target.

Activity Log

You can view the activity log on the Activity Log page. The activity log includes log entries for all jobs in the organization that have completed. The Activity Log page refreshes job details every 5 seconds.

Informatica Cloud clears older activity log entries when they exceed the maximum number of log entries configured for the organization. The Informatica Cloud administrator can also clear all activity log entries when viewing or editing the organization properties.
The Activity Log page displays the following information for each job:

<table>
<thead>
<tr>
<th>Activity Log Column</th>
<th>Description</th>
</tr>
</thead>
</table>
| Status              | Status of the job:  
- Success. The job completed without errors.  
- Warning. The job completed with errors.  
- Error. The job failed to complete. |
| Actions             | Actions that you can perform for the job, such as viewing a snapshot of the error log. |
| Name                | Name of the task. |
| Type                | Type of task. |
| End Time            | Date and time that the task completed. Dates and times appear in the following format: MM/DD/YY HH:MM:SS |
| Results             | Results of the job:  
- Success rows. Rows that were written to the target.  
- Error rows. Rows that were not written to the target. |

Viewing the Activity Log

The activity log displays log entries for all completed jobs.

1. Click **Home > Activity Log**.
2. Click a task link to show the details about the task.

**RELATED TOPICS:**
- “Viewing Organization Properties” on page 8
- “Error Log Files” on page 163

Viewing Activity Log Details

In the activity log, you can view the job details about each task that completed.

1. Click the task name link on the Activity Log page.
   
The Activity Log Details page appears and shows the following job details:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Name</td>
<td>Name of the task.</td>
</tr>
<tr>
<td>Task Type</td>
<td>Type of task.</td>
</tr>
</tbody>
</table>
| Status           | Status of the job. Valid values are:  
- Success. The job completed without errors.  
- Warning. The job completed with errors.  
- Error. The job failed to complete. |
| Download Logs    | Data loader and data synchronization tasks only. Allows you to download the session log file for a job with error rows. |
### Field Description

You must have the Admin role to download a session log.

Start Time | Date and time that the job started. Dates and times appear in the following format: MM/DD/YY HH:MM:SS. For example, 8/6/09 11:29:57 AM.
---|---
End Time | Date and time that the job completed. Dates and times appear in the following format: MM/DD/ YY HH:MM:SS.
Duration | The time it took to complete the job.
Source Success Rows | Number of rows that were successfully read from the source.
Source Error Rows | Number of rows that were not read from the source.
Target Success Rows | Number of rows that were successfully written to the target.
Target Error Rows | Number of rows that were not written to the target.
Transformation Errors | For data loader and data synchronization tasks: The total number of transformation errors. The maximum number of errors is 1 because the task fails after the first transformation error. For data replication tasks: The total number of transformation errors for all objects that are replicated.
View Error Logs | Data loader and data synchronization tasks only. A snapshot of the error log file. You can view the first 25 rows and 50 columns of the error log file when the job generates errors.

For data replication tasks, the Activity Log Details page also shows the following Activity Log Entries:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Status    | Status of the job:  
- Success. The job completed without errors.  
- Warning. The job completed with errors.  
- Error. The job failed to complete. |
| Actions   | Actions that you can perform for the job, such as viewing a snapshot of the error log. |
| Name      | Name of the task flow or target object. |
| Type      | Type of task, if applicable. |
| Start Time| Date and time that the task started. Dates and times appear in the following format: MM/DD/YY HH:MM:SS. For example, 8/6/09 11:29:57 AM. |
| End Time  | Date and time that the task completed. Dates and times appear in the following format: MM/DD/YY HH:MM:SS. |
| Success Rows | Total number of rows successfully read from the source and written to the target. Place the cursor over the value to see the number of successful source rows and successful target rows separately. |
| Error Rows | Total number of rows that were not successfully read from the source or written to the target. Place the cursor over the value to see the number of error source rows and error target rows separately. |
| Error Message | Shows the error message for each target object. |
2. Click the task name link to view details about the task.
   If you view the details about a task, click Done to return to the Activity Log Details page.

3. Click Done to return to the Activity Log page.

**Related Topics:**
- "Error Log Files" on page 163

### Searching the Activity Log Based on Task Name

You can search the activity log by task name to find jobs that completed successfully and failed.

1. In the Activity Log, enter all or part of the task name in the **Task Name Field**.
2. Click **Search**.
3. Click **Reset** to clear the search criteria.

### Searching the Activity Log Based on Activity Log Details

You can search the activity log by job details to find jobs that completed successfully and failed.

1. In the Activity Log, click **More Options**.
2. Enter the following search criteria:

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Name</td>
<td>Enter the task name.</td>
</tr>
</tbody>
</table>
| Status         | Select one of the following options:  
- Success. The job completed without errors.  
- Warning. The job completed with errors.  
- Error. The job failed to complete. |
| End Time       | Enter the start and end times of the job. Enter times in the following format: `YYYY-MM-DD HH:MM:SS`.  
If you enter the date and no time, Informatica Cloud uses 00:00:00 as the time.  
If you do not specify the end time, Informatica Cloud adds one day to the start time to determine the end time.  
You can also click the Calendar icon to select a date and use the current time. |
| Show Only      | Select one or more of the following options:  
- Non Zero Success Rows. Finds jobs that load at least one successful row.  
- Non Zero Error Rows. Finds jobs that load at least one error row. |
| Error Message  | Specify part or all of the error message string that appears in Results column. |

3. Click **Search**.
4. Click **Reset** to clear the search criteria.
5. Click a column header to sort search results by the column.
Activity Monitor

You can view details about tasks and task flows that are currently running in the activity monitor. You can view activity monitor information in a list view or detail view on the Activity Monitor page. If necessary, you can also stop a running job on the Activity Monitor page.

The activity monitor refreshes job details every 5 seconds.

Stopping a Job

In the activity monitor, you can stop a job that is running.

To stop a running job:
1. Click Home > Activity Monitor.
2. On the Activity Monitor, click the Stop Job icon next to the task or task flow that you want to stop.
   Informatica Cloud stops the task or task flow. You can view details about the stopped job on the activity log.

Viewing the Activity Monitor in List View

You can view the activity monitor in list view for all jobs in the organization that are currently running. Display the activity monitor in list view to get the task name, task type, and start time of each job that is currently running.

To view the activity monitor in list view:
1. Click Home > Activity Monitor.
   The activity monitor appears in list view on the Activity Monitor page by default.

Viewing the Activity Monitor in Detail View

You can view the activity monitor in detail view for all jobs in the organization that are currently running. Display the activity monitor in detail view to get the task name, task type, start time, end time, number of rows processed. The results in the detail view appear at the job level and object level. The activity monitor displays the source object for data replication jobs and shows target objects for data synchronization and data loader jobs. The activity monitor does not display object-level results for PowerCenter tasks.

For example, the detail view of the activity monitor shows the number of loaded records from each Salesforce source object, Account and Contact, in a data replication task that replicates Salesforce Account and Contact data to an Oracle database.

To view the activity monitor in detail view:
1. Click Home > Activity Monitor.
2. On the Activity Monitor page, click Detail.

Dashboard

The dashboard can display a scorecard for each data assessment task that previously ran. A scorecard summarizes the results of quality metrics included in the data assessment task.

Note: If you do not configure a data assessment task to show results on the dashboard, the dashboard is empty and displays links to access Informatica Cloud Services.
Configuring the Dashboard

You can configure the dashboard to show scorecards for data assessment tasks.

Note: You can also configure a scorecard to appear in the dashboard on the Schedule page of the Data Assessment Task Wizard.

1. Click Home > Dashboard.
2. On the Dashboard page, click Configure.
3. In the Configure Dashboard page, add the data assessment tasks to the Selected Items box to view the corresponding scorecards in the dashboard.
   - Data assessment tasks appear in the Selected Items box if you configured the data assessment task to show the results in the dashboard when you created the plan.
4. Remove a data assessment task from the Selected Items box to remove the corresponding scorecard from the dashboard.
5. Click OK to save your settings.

Viewing the Dashboard

You can view the scorecards for all data assessment tasks that previously ran on the dashboard. Click View Details for a particular scorecard to see more details about the scorecard.

To view the dashboard, click Home > Dashboard. The dashboards appear on the Dashboard page. The dashboard shows Salesforce fields, not the fields defined in data assessment tasks.

Data Files

The Data Files page shows all previously downloaded data files. Data files contain data assessment task analysis information. When you analyze a data assessment task, you can download the data quality results to a CSV data file. To display data of multiple languages, the Data Assessment service generates each CSV data file in the UTF-8 character set. You can use Microsoft Notepad or Excel to open a UTF-8 file.

The Data Assessment service deletes data files after 7 days. Save a data file to a local machine to retain the file for a longer period of time.

To view a data file:

1. Click Home > Data Files.
2. On the Data Files page, click the Download icon for the data file.
3. Click the Delete icon next to a data file to delete the data file.

Error Log Files

Informatica Cloud generates error log files for data loader and data synchronization tasks. An error log file shows rows that failed and the reason why each row failed.
For example, the following error appears in the error log file when the task tries to insert two records with the same external ID into a Salesforce target:

```
Error loading into target [HouseholdProduct__c] : Error received from salesforce.com. Fields [ExternalId__c]. Status code [DUPLICATE_VALUE]. Message [Duplicate external id specified: 1.0].
```

### Viewing a Snapshot of the Error Log File

You can view the first 25 rows and first 50 columns of the error log file through the Activity Log.

1. Click **Home > Activity Log**.
2. In the Actions column, click the **View Error Logs** icon for the job that you want to review.

### Viewing the Error Log File

You can view the entire error log file for a job. Error log files are saved in the installation directory of the Secure Agent that runs the task. You can find error log files in the following directory:

```
<SecureAgent_InstallDirectory>/main/rdtmDir/error
```

Use the start time of the job to identify the corresponding error log file. The timestamp of the error log file will be similar to the start time of the job.

If the error log contains Unicode data from the source, open the error log file in an application that can display Unicode characters.
Troubleshooting Overview

Use the following sections to troubleshoot errors in Informatica Cloud.

Troubleshooting a Secure Agent

I installed the Secure Agent, but I want to install another on a different machine. How do I do that?

On the new machine, use your login to connect to Informatica Cloud. Then, download and install the Secure Agent.

**RELATED TOPICS:**
- "Rules and Guidelines for the Secure Agent" on page 32
- “Downloading the Secure Agent Installation Wizard” on page 33
- “Installing the Secure Agent on Windows” on page 33
Secure Agent Errors

I started the Secure Agent, but the status is inactive.

The Secure Agent may take a few minutes to start. The status refreshes every 5 seconds. If the Secure Agent does not become active, complete the following tasks:

- If your organization uses a proxy server to access the internet, verify that the proxy settings are set correctly in Informatica Cloud. For more information, see Configuring the Proxy Settings.
- View the details in infaagent.log in the directory where you installed the Secure Agent.

The Secure Agent did not install or start correctly.

If the Secure Agent does not install or start correctly, complete the following tasks:
1. View the installation details in infaagent.log in the directory where you installed the Secure Agent.
2. View the application logs in the Event Viewer for a Secure Agent that runs on Windows.

Troubleshooting a Connection

Review the rules and guidelines for the connections.

RELATED TOPICS:

- “Rules and Guidelines for Connections” on page 44
- “Rules and Guidelines for Data Assessment Sources” on page 72
- “Rules and Guidelines for Data Loader Sources and Targets” on page 82
- “Rules and Guidelines for Data Replication Sources and Targets” on page 92
- “Rules and Guidelines for Data Synchronization Sources and Targets” on page 111
- “Rules and Guidelines for PowerCenter Sources and Targets” on page 134

Connection Errors

The connection fails to connect to a Salesforce account.

You may have to enter a Salesforce security token in the Salesforce connection details. If the security token is required and the Security Token field in the Salesforce connection is empty or invalid, the following error message appears when you test or create the connection:

    The login to Salesforce.com failed with the following message - LOGIN_MUST_USE_SECURITY_TOKEN:

Go to the Salesforce web site to obtain the security token. To avoid adding the security token to connection details, you can also add Informatica Cloud IP addresses to Trusted IP Ranges in your Salesforce account. For more information, see “Security Tokens and Trusted IP Ranges” on page 43. You can find additional information with in the Informatica How-To Library article: Activating IP Addresses for Salesforce Connections.

The connection I want to use does not display in the task wizard.
You do not have read permission for the connection. You need read permission on all Informatica Cloud objects that you want to use in a task or task flow. Ask your organization administrator for the appropriate permissions.

**Troubleshooting a Task**

The activity monitor and activity log show the number of rows that failed to load into the target. For more information about the reason for each failure, view the error log file.

You can find the error log files in the following directory:

```
<SecureAgent_InstallDirectory>\main\rdtmDir\error
```

If a task or task flow does not complete, you can click the Stop Job option in the activity monitor.

**RELATED TOPICS:**
- “Activity Monitor” on page 162
- “Activity Log” on page 158
- “Error Log Files” on page 163

**Task Errors**

I need to review details for a task, but I can't find the task anywhere!

You do not have read permission for the task. You need read permission to view task details. Ask your organization administrator for the appropriate permissions.

I can view task details, but I cannot edit a task.

You do not have update permission for the task. To edit a task, you need read and update permission for the task, and you need read permission for all objects used in the task. Ask your organization administrator for the appropriate permissions.

When I edit a task, one of the connections displays a blank field instead of the connection name.

You do not have read permission for the connection. To edit a task, you should have read permission for all objects used in the task. To avoid unexpected results, cancel all changes and ask your organization administrator for the appropriate permissions.

After you cancel out of the wizard, you can view the task details to determine the names of the objects for which you need permissions.

When creating a task, the connection that I want to use does not appear in the list of available connections.

You do not have read permission for the connection. You need read permission on all Informatica Cloud objects that you want to use in a task or task flow. Ask your organization administrator for the appropriate permissions.

The following error appears:

```
Unknown SQL datatype for field [Object name.Field name]. [Number that represents the datatype]
```

Informatica Cloud does not recognize the datatype of a source or target field.
Troubleshooting a Data Assessment Task

Review the rules and guidelines for the components of data assessment task.

**RELATED TOPICS:**
- “Rules and Guidelines for Data Assessment Sources” on page 72
- “Rules and Guidelines for Data Filters in Data Assessment Tasks” on page 74
- “Rules and Guidelines for Weights” on page 76
- “Rules and Guidelines for Field Mappings” on page 77
- “Rules and Guidelines for Data Filters” on page 147

Data Assessment Task Errors

The data file is missing from the Data Files page.

The Data Assessment service deletes data files after 7 days.

The Top Issues area overestimates the results for quality metrics.

The Top Issues area rounds percentages up to the nearest percent.

I cannot add a quality metric for a particular Salesforce object.

Some quality metrics are not applicable for all Salesforce objects. For example, you cannot include the Address Validation quality metric in a data assessment task for the Opportunity object. The Opportunity object does not contain address fields.

I do not see the trend chart in my scorecard.

You must run the data assessment task at least two times to see results in the trend chart.

Troubleshooting a Data Loader Task

Review the rules and guidelines for the components of a data loader task.

**RELATED TOPICS:**
- “Rules and Guidelines for Datatypes” on page 177
- “Rules and Guidelines for Data Loader Sources and Targets” on page 82
- “Rules and Guidelines for Running a Data Loader Task” on page 89
- “Rules and Guidelines for Data Filters” on page 147

Data Loader Task Errors

The data loader task fails.
A data loader task might fail for the following reasons:

- You tried to run a data loader task that was already running. You can run one instance of a data loader task at a time.
- You tried to run a data loader task, but the corresponding Secure Agent is not running. Verify that the Secure Agent is running.
- You tried to run a data loader task that reads data from or writes data to a database, but the corresponding database server is not running. Verify that the database server is running.
- You tried to run a data loader task that contains an invalid field mapping.

If the source or target changed after you created the field mapping for a data loader task and you try to run the data loader task, the following error appears:

```plaintext
TE_7002 Transformation Parse Fatal Error; transformation stopped...
```

The data loader task fails in the following cases:

- A required field is not mapped in the field mapping. You may have changed the default field mappings or a required field may have been added to the source or target after you created the data loader task. You must edit the data loader task and include all required fields before you can successfully run the data loader task.
- A field in a field mapping no longer exists in the source or target. The structure of the source or target can change after you create the field mapping in the data loader task. Resolve the field mappings in the data loader task that point to nonexistent fields.
- You tried to run a data synchronization task that contains an invalid field name. When a field name is invalid, the task fails with the following error:

```plaintext
```

- You tried to run a data loader task and the field in the data filter changed in one of the following ways:
  - The field no longer exists in the source.
  - The datatype of the field changed.

  Edit the data filters in the data loader task definition based on the latest version of the source.
- You tried to run a task that writes data from a flat file to a Salesforce object, but one of the external ID fields in the flat file is not found in the related Salesforce object. When this occurs, the task fails with following error:

```plaintext
[Salesforce object name] : Error received from salesforce.com. Fields []). Status code [INVALID_FIELD]. Message [Foreign key external ID: not found for field <field name> in entity <source Object>].
```

To resolve, verify that the external ID values exist in Salesforce.

This error also occurs when the external ID field in the flat file contains a blank space (" "). To resolve, remove the blank space from the external ID field, and then run the task again.
- You tried to run a data loader task with a flat file source and previously saved the file as CSV in Excel. When you save a CSV file in Excel, by default, Excel changes the date format to MM/dd/yyyy HH:mm. To prevent the data loader task from failing, perform one of the following steps:
  - Verify that the date format in the Flat File connection is correct based on the format used in the CSV file.
  - If the format in the connection is incorrect, specify the correct date format.
  - If a field does not have the correct date format, use an expression in the field mapping to reformat the date field to the format expected by the target in the task.
- You imported a data loader process from Salesforce in to Informatica Cloud, and then ran the data loader task. To resolve this issue, click Edit, and then Save to save the data loader task.

The data preview does not show the latest changes to the dataset for the data loader task.
The Data Loader Task Wizard caches the dataset for connections when you configure the task. The Data Loader Task Wizard deletes the cache after you save the task and refreshes the cache each time you configure the task. To refresh the cache, save or cancel the task, and then edit the task.

An error appears when you validate the field mapping after importing a data loader process.

After you import a data loader process and validate the field mapping in the task, the following error may appear:

Missing data type on one or more source fields. Please click Refresh Fields button to reload fields.

To resolve this issue, refresh the fields. When you refresh the fields, the service retrieves the latest field attributes, including the field datatypes.

Troubleshooting a Data Replication Task

Review the rules and guidelines for the components of a data replication task.

**RELATED TOPICS:**
- “Rules and Guidelines for Datatypes” on page 177
- “Rules and Guidelines for Data Replication Sources and Targets” on page 92
- “Rules and Guidelines for Configuring Data Replication Tasks” on page 94
- “Rules and Guidelines for Running a Data Replication Task” on page 100
- “Rules and Guidelines for Running Incremental Loads” on page 106
- “Rules and Guidelines for Data Filters” on page 147

Data Replication Task Errors

I chose a source connection while creating a data replication task, but I cannot see any available objects to replicate.

When you choose a source connection in the data replication task, Informatica Cloud connects to the source to retrieve objects that it can replicate. If the connection is not valid or the connection does not have privilege to view the object in the source system, Informatica Cloud cannot connect to the source and shows no objects.

I cannot find the Salesforce Vote or UserProfileFeed objects in the list of available objects to replicate. Also, I cannot run an existing task that includes those objects.

The Data Replication service does not replicate the Salesforce Vote or UserProfileFeed objects.

If you have a data replication task from a previous version of Informatica Cloud that includes the Vote or UserProfileFeed objects, the Data Replication service does not run the task. To enable the task to run, remove the Vote or UserProfileFeed objects from the task.

An error message appears while I configure the target for the data replication task.

You cannot configure multiple data replication tasks to replicate the same source object to the same target object. For example, you cannot configure two data replication tasks to write Salesforce Account data to the SF_ACCOUNT Oracle database table. When you select the same target connection and target object that is used by another data replication task, an error appears.
If you replicate all objects for the source connection, the following error appears:

Duplicate data replication task. The data replication task <name of the other data replication task> is also replicating the same objects from the same source to the same target with the same target prefix as this data replication task. Change the target or enter a different target prefix.

If you replicate specific objects for the source connection, the following error appears:

Duplicate data replication task for <object name> object with a target prefix of <target prefix>. The data replication task <name of the other data replication task> is already replicating an object whose target prefixed name is same as this one. Change the target connection or enter a different target prefix or remove the object from the task.

To write data from sources with the same name to different target objects in the same target connection, configure different target prefixes. For more information, see “Target Prefixes” on page 95.

I ran a data replication task, but I did not get results in the expected time period.

A data replication task may take longer than expected for the following reasons:

- The data replication task is in queue because too many data replication tasks are running simultaneously. Contact Informatica Global Customer Support to check the status of the data replication task.
- You have a large amount of data in the source.
- Salesforce or database servers have an outage or are experiencing heavy traffic.

If a data replication task replicates all Salesforce objects and tries to write a null value to a numeric field of an Oracle database, the database driver issues the following warning message:


Informatica Cloud displays this warning in the activity log. This is not a fatal error. The Data Replication service writes the row to the target and the data replication job succeeds.

Why did my data replication task fail?

A data replication task might fail for the following reasons:

- You run an incremental load after the datatype, precision, or scale of a Salesforce field changed. To replicate Salesforce data after these changes, configure the data replication task to perform a full load.
- The source or target connection might have timed out or it might be down.
- The Secure Agent is not running.
- The target database server is not running. If you restart the database server, the test connection to the database succeeds, but the data replication task fails, restart the Secure Agent.
- Your organization’s license to use the Data Replication service expired.
- You tried to run a data replication task that was already running.

If you verify that the data replication task did not fail for the above reasons, contact Informatica Global Customer Support and provide the scenario and, if available, error messages.

I did not receive an email notification that stated that my data replication task failed.

You can configure the task to send an email to you if the session completes, completes with errors, or fails. If you should receive email for all tasks, ask the organization administrator to configure the organization properties to send you email notification.

The target table name is truncated.

If the length of the source object name exceeds the maximum number of characters allowed for a table name in a relational target, the Data Replication service truncates the corresponding table name. For more information, see “Truncating Table Names” on page 103.
The table in the target database does not contain the same data as the corresponding source object.

The Data Replication service overwrites the data in a target table when two source objects from different data replication tasks share the same target table. When multiple source objects write to the same target table, the Data Replication service overwrites table data each time you replicate one of the source objects.

To avoid overwriting tables, use a different target table name prefix for each data replication task. For more information, see “Duplicate Table Names from Different Data Replication Tasks” on page 103.

The column name in the target database is truncated.

If the length of the source object name exceeds the maximum number of characters, the Data Replication service truncates the corresponding column name based on the maximum number of characters allowed for the column name in the target database. For more information, see “Truncating Column Names” on page 103.

Numeric data is loaded incorrectly when I run a replication task that writes Oracle data to a flat file.

When a replication task writes Oracle data with the Number datatype to a flat file and the scale is 0, the number is rounded with a precision of 15 because the Data Replication service converts the Number datatype to Double (15). To prevent rounding, specify a scale other than 0 for the source datatype. When you specify a non-zero scale, no rounding occurs because the Data Replication service converts the Number datatype to Decimal.

A data overflow error occurs when I run a replication task that writes MySQL data to a flat file.

When a replication task writes MySQL data with BigInt datatype to a flat file and the data length is greater than 10, a data overflow error occurs because the Data Replication service converts BigInt to Int (10). To prevent the data overflow error, change the BigInt source datatype to Decimal.

The following error appears when I try to create the target for the data replication task:

Please use Reset Target first and then try Create Target again.

This message appears for the following reasons:

♦ You click the Create Target option twice.
♦ You edit the data replication task, and then click Create Target.

To resolve, click Reset Target, and then click Create Target.

Troubleshooting a Data Synchronization Task

Review the rules and guidelines for the components of a data synchronization task.

RELATED TOPICS:

♦ “Rules and Guidelines for Datatypes” on page 177
♦ “Rules and Guidelines for Data Synchronization Sources and Targets” on page 111
♦ “Rules and Guidelines for a Multiple-Object Source” on page 110
♦ “Rules and Guidelines for Lookups” on page 114
♦ “Rules and Guidelines for Running a Data Synchronization Task” on page 129
♦ “Rules and Guidelines for Data Filters” on page 147
Data Synchronization Task Errors

The data synchronization task fails.

A data synchronization task might fail for the following reasons:

- You tried to run a data synchronization task that was already running. You can run one instance of a data synchronization task at a time.
- You tried to run a data synchronization task, but the corresponding Secure Agent is not running. Verify that the Secure Agent is running.
- You tried to run a data synchronization task that reads data from or writes data to a database, but the corresponding database server is not running. Verify that the database server is running.
- You tried to run a data synchronization task that contains an invalid field mapping.

If the source or target changed after you created the field mapping for a data synchronization task and you try to run the data synchronization task, the following error appears:

```
TE_7002 Transformation Parse Fatal Error; transformation stopped...
```

The data synchronization task fails in the following cases:

- A required field is not mapped in the field mapping. You may have changed the default field mappings or a required field may have been added to the source or target after you created the data synchronization task. You must edit the data synchronization task and include all required fields before you can successfully run the data synchronization task. For more information about required fields in field mappings of data synchronization tasks, see “Step 5. Configure the Field Mapping” on page 125.
- A field in a field mapping no longer exists in the source or target. The structure of the source or target can change after you create the field mapping in the data synchronization task. Resolve the field mappings in the data synchronization task that point to nonexistent fields.
- You tried to run a data synchronization task that contains an invalid field name. When a field name is invalid, the task fails with the following error:

```
```
- You tried to run a data synchronization task and the field in the data filter changed in one of the following ways:
  - The field no longer exists in the source.
  - The datatype of the field changed.

Edit the data filters in the data synchronization task definition based on the latest version of the source.
- You tried to run a task that writes data from a flat file to a Salesforce object, but one of the external ID fields in the flat file is not found in the related Salesforce object. When this occurs, the task fails with following error:

```
[Salesforce object name] : Error received from salesforce.com. Fields []. Status code [INVALID_FIELD]. Message [Foreign key external ID: not found for field <field name> in entity <source object>].
```

To resolve, verify that the external ID values exist in Salesforce.

This error also occurs when the external ID field in the flat file contains a blank space (" "). To resolve, remove the blank space from the external ID field, and then run the task again.

The data preview does not show the latest changes to the dataset for the data synchronization task.

The Data Synchronization Task Wizard caches the dataset for connections when you configure the task. The Data Synchronization Task Wizard deletes the cache after you save the task and refreshes the cache each time you configure the task. To refresh the cache, save or cancel the task, and then edit the task.

For FTP/SFTP connections, data preview shows the latest changes for the dataset when you update the local file to match the remote file.
I do not see the target when creating the task.

The list of targets are based on the task operation. For more information see "Task Operations" on page 46.

The data synchronization task inserted all data into the target, instead of performing the update/upsert/delete task operation configured for the task.

When you use an active plug-in with a data synchronization task that includes a custom source, the Data Synchronization service ignores the configured target option for the task and tries to insert data to the target.

Troubleshooting a PowerCenter Task

Review the rules and guidelines for the components of a PowerCenter task.

**RELATED TOPICS:**
- "Rules and Guidelines for PowerCenter Sources and Targets" on page 134
- "Rules and Guidelines for Configuring a PowerCenter Task" on page 135
- "Rules and Guidelines for Running a PowerCenter Task" on page 137

**PowerCenter Task Errors**

The following error appears when you import a workflow XML file:

The PowerCenter XML file is empty, please select a valid PowerCenter XML file.

The workflow XML file is empty.

The following error appears when you import a workflow XML file:

The PowerCenter file content type is not text/xml, please select a valid PowerCenter XML file.

The workflow XML file is not valid.

The following error appears when you run a PowerCenter task with more than 10 partitions:

TM_6710 Error: Unable to allocate storage for statistics for transformation <transformation name>.

The PowerCenter workflow cannot contain more than 10 partitions for sources and targets.

The following error appears when you run a PowerCenter task:

PowerCenter task <PowerCenter task name> failed to run. Another instance of the task is currently executing.

You cannot run multiple instances of a PowerCenter task simultaneously. If you configured the task to run on a schedule, increase the time interval between the scheduled tasks to prevent multiple instances of the task from running simultaneously. If you run the task manually, wait for the currently running instance of the task to complete before starting it again. You can view currently running tasks in the activity monitor.

The PowerCenter task does not complete.

PowerCenter tasks run by a Secure Agent on Windows 7 (64 bit) might not complete. To resolve the issue, configure a network login for the Secure Agent service.
Troubleshooting Field Mappings

Review the rules and guidelines for the field mappings.

**RELATED TOPICS:**

- “Rules and Guidelines for Validating Field Mappings” on page 141
- “Rules and Guidelines for Expressions” on page 143
- “Rules and Guidelines for Data Filters” on page 147

Troubleshooting Scheduled Tasks

The task does not run at the scheduled time.

A task does not run at the scheduled time if another instance of it is already running when the schedule tries to start the task. For example, you schedule a task to run every 5 minutes. The first task starts at 12 p.m., but does not complete until 12:06 p.m. The second instance of the task does not run at 12:05 p.m. because the first instance has not completed. Informatica Cloud starts the next task at 12:10 p.m.

To resolve this issue, change the schedule to allow the task to complete before starting the next task run.

Troubleshooting Monitoring Jobs

The task does not show up on the activity monitor or activity log.

If a task or task flow is running, it displays on the activity monitor. If it has completed, it displays on the activity log.

If a task or task flow does not display on either the activity monitor or activity log, you might not have the appropriate permission to view job details. You need read and run permission on a task to view job details on the activity log or activity monitor. Ask your organization administrator if you have the appropriate permissions.

One of the tasks in a task flow does not display in the Activity Monitor when it should be running. It also does not display in the Activity Log Details, even though the rest of the tasks in the task flow have completed.

You do not have read permission for the missing task. When you run a task flow, Informatica Cloud runs all tasks in the task flow, but only the tasks for which you have read permission display in the Activity Monitor and the Activity Log Details. The final row counts for the task flow include all tasks.

One of the tasks in a schedule does not display in the Activity Monitor when it should be running. It also does not show up in the Activity Log, even though the rest of the tasks in the schedule have completed.

You do not have read permission for the missing task. Informatica Cloud runs all tasks associated with a schedule, but only the tasks for which you have read permission display in the Activity Monitor and the Activity Log.
Troubleshooting Security

When I try to log in to Informatica Cloud, I receive the following error message:

Login Failed. Access from your IP address is denied.

Your organization uses two-factor authentication. This requires you to log in from within the specified IP address ranges. For help logging in from an appropriate IP address, contact your Informatica Cloud administrator.

If the Informatica Cloud administrator cannot log in because an incorrect IP address range has been configured for the organization, contact Informatica Global Customer Support.

I received the following security violation error:

There may have been a security violation while accessing the site. Verify that there are no malicious scripts running in your browser. This error also appears when you submit the form multiple times through a browser reload.

This error appears when you click an option on a page while the page is still loading from a previous click. Click the Here link to return to Informatica Cloud.

When I try to view the details about an object, such as a connection or data replication task, the Object Not Found page displays.

The object was recently deleted. The Object Not Found page appears when an object no longer exists. Refresh the page to display current objects.

When I try to perform a task, the Access Denied page displays.

The Access Denied page displays when you try to perform a task that is not allowed for your user account. You might not have the role, user group, or object-level permissions to perform the task. If you need access to perform the task, ask the Informatica Cloud Administrator to review your user account.
Datatype Reference Overview

When the Data Replication service replicates source data, it converts the source datatype to a compatible datatype in the target database.

Informatica Cloud converts datatypes to and from the following sources and targets:

- Microsoft SQL Server 2000
- Microsoft SQL Server 2005
- Microsoft SQL Server 2008
- MySQL
- Oracle
- Salesforce

Rules and Guidelines for Datatypes

Use the following rules and guidelines to for datatypes and conversion:

- The task may have unexpected results if you map incompatible datatypes between source and target fields or between the output of an expression and a target field. For example, if you map a datetime column of a MySQL database source to an integer column of a Salesforce target in a data loader task, the data loader task fails.
A data loader or data synchronization task fails when you map source fields of the following datatype and database type to a Salesforce target:

<table>
<thead>
<tr>
<th>Datatype</th>
<th>Database Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>tinyint MySQL</td>
<td></td>
</tr>
<tr>
<td>tinyint SQL Server</td>
<td></td>
</tr>
<tr>
<td>interval year to month</td>
<td>Oracle</td>
</tr>
<tr>
<td>interval day to second</td>
<td>Oracle</td>
</tr>
<tr>
<td>rowid Oracle</td>
<td></td>
</tr>
<tr>
<td>urowid Oracle</td>
<td></td>
</tr>
</tbody>
</table>

In most cases, the Data Replication service creates the same precision and scale in the target as the source. In other cases, the Data Replication service uses a different precision or scale.

- In some cases, the Data Replication service does not specify the precision or scale, and the database uses the defaults. For example, the source is MySQL and source datatype is Double(12). The Data Replication service creates the Number datatype in an Oracle target and does not specify the precision.

- If the source is not Oracle, the target is Oracle, and the source datatype is nchar, nvarchar, or ncllob, the service multiples the source field precision by 2, up to a maximum of 64000, to obtain the target field precision.

- If the source is MySQL, the target is Microsoft SQL Server 2000 or 2005, and the source datatype is date or time, the target field datatype is Timestamp(23, 3).

- Datatypes do not exist for flat files.

- If a data replication task writes data from MySQL to a flat file and the source contains time data, the Data Replication service converts the time data to a date/time datatype, where the date is the current date and the time is the time specified in the source. You can use a string function in an expression to remove the date before loading the flat file.

- For an Oracle source or target, the precision of a Number field must be greater than or equal to the scale. Otherwise, the task fails.

- When a data loader or data synchronization task writes 17-digit or 18-digit numeric data with no scale from Salesforce to an Oracle column with a Number datatype, the task may produce unexpected output in the target. For example, the data loader task writes the Salesforce value 67890123456789045 as 67890123456789048 in an Oracle target.

- When you use an ODBC connection for an Oracle database target, ensure that the maximum precision for an Oracle table column does not exceed the following values: char(1999), varchar(3999), nvarchar(3998), and nchar(3998).

- The task may load corrupt data into the target if the data comes from a source field of the Real datatype from Microsoft SQL Server.
The following table shows the conversion of Microsoft SQL Server 2000 datatypes to all target datatypes:

<table>
<thead>
<tr>
<th>Microsoft SQL Server 2000 Source</th>
<th>Flat File Target</th>
<th>Microsoft SQL Server Target</th>
<th>MySQL Target</th>
<th>Oracle Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigint</td>
<td>Int</td>
<td>Bigint</td>
<td>Bigint</td>
<td>Number</td>
</tr>
<tr>
<td>Binary</td>
<td>Not supported.</td>
<td>Binary</td>
<td>Binary</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Bit</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Char</td>
<td>Nstring</td>
<td>Char</td>
<td>Char if precision is 255 or less.</td>
<td>Char</td>
</tr>
<tr>
<td>Cursor</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Datetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Date</td>
</tr>
<tr>
<td>Decimal</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number</td>
</tr>
<tr>
<td>Float</td>
<td>Number</td>
<td>Float</td>
<td>Double</td>
<td>Number</td>
</tr>
<tr>
<td>Image</td>
<td>Not supported.</td>
<td>Image</td>
<td>Longblob</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Money</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number</td>
</tr>
<tr>
<td>Nchar</td>
<td>Nstring</td>
<td>Nchar</td>
<td>Char if precision is 255 or less.</td>
<td>Nchar</td>
</tr>
<tr>
<td>Ntext</td>
<td>Nstring</td>
<td>Ntext</td>
<td>Text</td>
<td>Nclob</td>
</tr>
<tr>
<td>Numeric</td>
<td>Number</td>
<td>Numeric</td>
<td>Double</td>
<td>Number</td>
</tr>
<tr>
<td>NVarchar</td>
<td>Nstring</td>
<td>NVarchar</td>
<td>Varchar if precision is 255 or less.</td>
<td>NVarchar2</td>
</tr>
<tr>
<td>Real</td>
<td>Number</td>
<td>Real</td>
<td>Double</td>
<td>Number</td>
</tr>
<tr>
<td>Smalldatetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Date</td>
</tr>
<tr>
<td>Smallint</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Smallmoney</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number</td>
</tr>
<tr>
<td>Sql_variant</td>
<td>Nstring</td>
<td>Varchar(255)</td>
<td>Varchar(255)</td>
<td>Varchar2(255)</td>
</tr>
<tr>
<td>Table</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
</tbody>
</table>
### Datatype Conversion from Microsoft SQL Server 2005

The following table shows the conversion of Microsoft SQL Server 2005 datatypes to all target datatypes:

#### Table 6. Datatype Conversions From Microsoft SQL Server 2005

<table>
<thead>
<tr>
<th>Microsoft SQL Server 2005 Source</th>
<th>Flat File Target</th>
<th>Microsoft SQL Server Target</th>
<th>MySQL Target</th>
<th>Oracle Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Nstring</td>
<td>Text</td>
<td>Text</td>
<td>Clob</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Not supported.</td>
<td>Binary</td>
<td>Binary</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Tinyint</td>
<td>Int</td>
<td>Tinyint</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Uniqueidentifier</td>
<td>Nstring</td>
<td>Char(36)</td>
<td>Char(36)</td>
<td>Char(36)</td>
</tr>
<tr>
<td>Varbinary</td>
<td>Not supported.</td>
<td>Varbinary</td>
<td>Varbinary</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Varchar</td>
<td>Nstring</td>
<td>Varchar</td>
<td>Varchar if precision is 255 or less. Text if precision is greater than 255.</td>
<td>Varchar</td>
</tr>
</tbody>
</table>

<p>| Bigint                           | Int              | Bigint                       | Bigint       | Number       |
| Binary                           | Not supported.   | Binary                       | Binary       | Not supported.|
| Bit                              | Int              | Int                          | Int          | Number       |
| Char                             | Nstring          | Char                         | Char if precision is 255 or less. Text if precision is greater than 255. | Char |
| Cursor                           | Not supported.   | Not supported.               | Not supported. | Not supported.|
| Datetime                         | Datetime         | Datetime                     | Datetime     | Date         |
| Decimal                          | Number           | Decimal                      | Decimal      | Number       |
| Float                            | Number           | Float                        | Double       | Number       |
| Image                            | Not supported.   | Image                        | Longblob     | Not supported.|
| Int                              | Int              | Int                          | Int          | Number       |
| Money                            | Number           | Decimal                      | Decimal      | Number       |</p>
<table>
<thead>
<tr>
<th>Microsoft SQL Server 2005 Source</th>
<th>Flat File Target</th>
<th>Microsoft SQL Server Target</th>
<th>MySQL Target</th>
<th>Oracle Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nchar</td>
<td>Nstring</td>
<td>Nchar</td>
<td>Char if precision is 255 or less. Text if precision is greater than 255.</td>
<td>Nchar</td>
</tr>
<tr>
<td>Ntext</td>
<td>Nstring</td>
<td>Ntext</td>
<td>Text</td>
<td>Nclob</td>
</tr>
<tr>
<td>Numeric</td>
<td>Number</td>
<td>Numeric</td>
<td>Double</td>
<td>Number</td>
</tr>
<tr>
<td>Nvarchar</td>
<td>Nstring</td>
<td>Nvarchar</td>
<td>Varchar if precision is 255 or less. Text if precision is greater than 255.</td>
<td>Nvarchar2</td>
</tr>
<tr>
<td>Real</td>
<td>Number</td>
<td>Real</td>
<td>Float</td>
<td>Number</td>
</tr>
<tr>
<td>Smalldatetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Date</td>
</tr>
<tr>
<td>Smallint</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Smallmoney</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number</td>
</tr>
<tr>
<td>Sql_variant</td>
<td>Nstring</td>
<td>Varchar(255)</td>
<td>Varchar(255)</td>
<td>Varchar2(255)</td>
</tr>
<tr>
<td>Table</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Text</td>
<td>Nstring</td>
<td>Text</td>
<td>Text</td>
<td>Clob</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Not supported.</td>
<td>Binary</td>
<td>Binary</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Tinyint</td>
<td>Int</td>
<td>Tinyint</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Uniqueidentifier</td>
<td>Nstring</td>
<td>Char(36)</td>
<td>Char(36)</td>
<td>Char(36)</td>
</tr>
<tr>
<td>Varbinary</td>
<td>Not supported.</td>
<td>Varbinary</td>
<td>Varbinary</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Varchar</td>
<td>Nstring</td>
<td>Varchar</td>
<td>Varchar if precision is 255 or less. Text if precision is greater than 255.</td>
<td>Varchar</td>
</tr>
<tr>
<td>Xml</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
</tbody>
</table>
Datatype Conversion from Microsoft SQL Server 2008

The following table shows the conversion of Microsoft SQL Server 2008 datatypes to all target datatypes:

Table 7. Datatype Conversions From Microsoft SQL Server 2008

<table>
<thead>
<tr>
<th>Microsoft SQL Server 2008 Source</th>
<th>Flat File Target</th>
<th>Microsoft SQL Server Target</th>
<th>MySQL Target</th>
<th>Oracle Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigint</td>
<td>Int</td>
<td>Bigint</td>
<td>Bigint</td>
<td>Number</td>
</tr>
<tr>
<td>Binary</td>
<td>Not supported.</td>
<td>Binary</td>
<td>Binary</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Bit</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Char</td>
<td>Nstring</td>
<td>Char</td>
<td>Char if precision is 255 or less. Text if precision is greater than 255.</td>
<td>Char</td>
</tr>
<tr>
<td>Cursor</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Date</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Datetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Date</td>
</tr>
<tr>
<td>Datetime2</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Datetimeoffset</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Decimal</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number</td>
</tr>
<tr>
<td>Float</td>
<td>Number</td>
<td>Float</td>
<td>Double</td>
<td>Number</td>
</tr>
<tr>
<td>Image</td>
<td>Not supported.</td>
<td>Image</td>
<td>Longblob</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Money</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number</td>
</tr>
<tr>
<td>Nchar</td>
<td>Nstring</td>
<td>Nchar</td>
<td>Char if precision is 255 or less. Text if precision is greater than 255.</td>
<td>Nchar</td>
</tr>
<tr>
<td>Ntext</td>
<td>Nstring</td>
<td>Ntext</td>
<td>Text</td>
<td>Nclob</td>
</tr>
<tr>
<td>Numeric</td>
<td>Number</td>
<td>Numeric</td>
<td>Double</td>
<td>Number</td>
</tr>
<tr>
<td>Nvarchar</td>
<td>Nstring</td>
<td>Nvarchar</td>
<td>Varchar if precision is 255 or less. Text if precision is greater than 255.</td>
<td>Nvarchar2</td>
</tr>
<tr>
<td>Real</td>
<td>Number</td>
<td>Real</td>
<td>Float</td>
<td>Number</td>
</tr>
<tr>
<td>Smalldatetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Date</td>
</tr>
<tr>
<td>Smallint</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
</tbody>
</table>
### Datatype Conversion from MySQL

The following table shows the conversion of MySQL datatypes to all target datatypes:

#### Table 8. Datatype Conversions From MySQL

<table>
<thead>
<tr>
<th>MySQL Source</th>
<th>Flat File Target</th>
<th>Microsoft SQL Server Target</th>
<th>MySQL Target</th>
<th>Oracle Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigint</td>
<td>Int</td>
<td>Bigint</td>
<td>Bigint</td>
<td>Number</td>
</tr>
<tr>
<td>Binary</td>
<td>Not supported.</td>
<td>Binary</td>
<td>Binary</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Bit</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Blob</td>
<td>Not supported.</td>
<td>Image</td>
<td>Longblob</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Bool</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Boolean</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Char</td>
<td>Nstring</td>
<td>Char (Nchar if MySQL code page is UTF-8)</td>
<td>Char</td>
<td>Char (Nchar if MySQL code page is UTF-8)</td>
</tr>
<tr>
<td>Date</td>
<td>Datetime</td>
<td>Not supported.</td>
<td>Date</td>
<td>Date</td>
</tr>
<tr>
<td>MySQL Source</td>
<td>Flat File Target</td>
<td>Microsoft SQL Server Target</td>
<td>MySQL Target</td>
<td>Oracle Target</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------</td>
<td>-----------------------------</td>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Datetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Date</td>
</tr>
<tr>
<td>Dec</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number(p,s)</td>
</tr>
<tr>
<td>Decimal</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number(p,s)</td>
</tr>
<tr>
<td>Double Precision</td>
<td>Number</td>
<td>Float</td>
<td>Double(p)</td>
<td>Number</td>
</tr>
<tr>
<td>Double(p,s)</td>
<td>Number</td>
<td>Decimal</td>
<td>Double(p,s)</td>
<td>Number(p,s)</td>
</tr>
<tr>
<td>Double[(p)]</td>
<td>Number</td>
<td>Float</td>
<td>Double(p)</td>
<td>Number</td>
</tr>
<tr>
<td>Enum</td>
<td>Nstring</td>
<td>Char (Nchar if MySQL code page is UTF-8)</td>
<td>Char</td>
<td>Char (Nchar if MySQL code page is UTF-8)</td>
</tr>
<tr>
<td>Float(p,s)</td>
<td>Number</td>
<td>Decimal</td>
<td>Double(p,s)</td>
<td>Number</td>
</tr>
<tr>
<td>Float[(p)]</td>
<td>Number</td>
<td>Real, Float</td>
<td>Float(p)</td>
<td>Number</td>
</tr>
<tr>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Integer</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Longblob</td>
<td>Not supported.</td>
<td>Image</td>
<td>Longblob</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Longtext</td>
<td>Nstring</td>
<td>Text (Ntext if MySQL code page is UTF-8)</td>
<td>Text</td>
<td>Clob (Nclob if MySQL code page is UTF-8)</td>
</tr>
<tr>
<td>Mediumblob</td>
<td>Not supported.</td>
<td>Image</td>
<td>Longblob</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Mediumint</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Mediumtext</td>
<td>Nstring</td>
<td>Text (Ntext if MySQL code page is UTF-8)</td>
<td>Text</td>
<td>Clob (Nclob if MySQL code page is UTF-8)</td>
</tr>
<tr>
<td>Serial</td>
<td>Int</td>
<td>Bigint</td>
<td>Bigint</td>
<td>Number</td>
</tr>
<tr>
<td>Set</td>
<td>Nstring</td>
<td>Char (Nchar if MySQL code page is UTF-8)</td>
<td>Char</td>
<td>Char (Nchar if MySQL code page is UTF-8)</td>
</tr>
<tr>
<td>Smallint</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td>Text</td>
<td>Nstring</td>
<td>Text (Ntext if MySQL code page is UTF-8)</td>
<td>Text</td>
<td>Clob (Nclob if MySQL code page is UTF-8)</td>
</tr>
<tr>
<td>Time</td>
<td>Datetime</td>
<td>Not supported.</td>
<td>Time</td>
<td>Date*</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Date</td>
</tr>
<tr>
<td>Tinyblob</td>
<td>Not supported.</td>
<td>Binary</td>
<td>Binary</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Tinyint</td>
<td>Int</td>
<td>Int</td>
<td>Int</td>
<td>Number</td>
</tr>
<tr>
<td><strong>MySQL Source</strong></td>
<td><strong>Flat File Target</strong></td>
<td><strong>Microsoft SQL Server Target</strong></td>
<td><strong>MySQL Target</strong></td>
<td><strong>Oracle Target</strong></td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------</td>
<td>---------------------------------</td>
<td>------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Tinytext</td>
<td>Nstring</td>
<td>Varchar (Nvarchar if MySQL code page is UTF-8)</td>
<td>Varchar</td>
<td>Varchar2 (Nvarchar if MySQL code page is UTF-8)</td>
</tr>
<tr>
<td>Varbinary</td>
<td>Not supported.</td>
<td>Varbinary</td>
<td>Varbinary</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Varchar</td>
<td>Nstring</td>
<td>Varchar (Nvarchar if MySQL code page is UTF-8)</td>
<td>Varchar</td>
<td>Varchar2 (Nvarchar if MySQL code page is UTF-8)</td>
</tr>
<tr>
<td>Year</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
</tbody>
</table>

*The data replication task converts MySQL time data to Oracle date data because Oracle does not contain a time datatype. Oracle date data contains the date and time. The task writes the current date and the time specified in the MySQL source to the Oracle date field.*

---

**Datatype Conversion from Oracle**

The following table shows the conversion of Oracle datatypes to all target datatypes:

**Table 9. Datatype Conversion From Oracle**

<table>
<thead>
<tr>
<th><strong>Oracle Source</strong></th>
<th><strong>Flat File Target</strong></th>
<th><strong>Microsoft SQL Server Target</strong></th>
<th><strong>MySQL Target</strong></th>
<th><strong>Oracle Target</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bfile</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Binary_double</td>
<td>Number</td>
<td>Float</td>
<td>Double</td>
<td>Number</td>
</tr>
<tr>
<td>Blob</td>
<td>Not supported.</td>
<td>Image</td>
<td>Blob</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Char</td>
<td>Nstring</td>
<td>Char</td>
<td>Char if precision is 255 or less. Text if precision is greater than 255.</td>
<td>Char</td>
</tr>
<tr>
<td>Clob</td>
<td>Nstring</td>
<td>Text</td>
<td>Text</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Date</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Date</td>
</tr>
<tr>
<td>Dec</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number</td>
</tr>
<tr>
<td>Decimal</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number</td>
</tr>
<tr>
<td>Double Precision</td>
<td>Number</td>
<td>Float</td>
<td>Double</td>
<td>Number</td>
</tr>
<tr>
<td>Float</td>
<td>Number</td>
<td>Float</td>
<td>Double</td>
<td>Number</td>
</tr>
<tr>
<td>Int</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number</td>
</tr>
<tr>
<td>Oracle Source</td>
<td>Flat File Target</td>
<td>Microsoft SQL Server Target</td>
<td>MySQL Target</td>
<td>Oracle Target</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------</td>
<td>-----------------------------</td>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Integer</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number</td>
</tr>
<tr>
<td>Interval Year To Month</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Long</td>
<td>Nstring</td>
<td>Text</td>
<td>Text</td>
<td>Clob</td>
</tr>
<tr>
<td>Nchar</td>
<td>Nstring</td>
<td>Nchar</td>
<td>Char if precision is 255 or less. Text if precision is greater than 255.</td>
<td>Nchar</td>
</tr>
<tr>
<td>Nclob</td>
<td>Nstring</td>
<td>Ntext</td>
<td>Text</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Number(p,s)</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number</td>
</tr>
<tr>
<td>Number[(p)]</td>
<td>Number</td>
<td>Float</td>
<td>Float</td>
<td>Number</td>
</tr>
<tr>
<td>Numeric</td>
<td>Number</td>
<td>Float</td>
<td>Float</td>
<td>Number</td>
</tr>
<tr>
<td>Nvarchar2</td>
<td>Nstring</td>
<td>Nvarchar</td>
<td>Varchar if precision is 255 or less. Text if precision is greater than 255.</td>
<td>Nvarchar2</td>
</tr>
<tr>
<td>Raw</td>
<td>Not supported.</td>
<td>Varbinary</td>
<td>Varbinary</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Real</td>
<td>Number</td>
<td>Float</td>
<td>Double</td>
<td>Number</td>
</tr>
<tr>
<td>Rowid</td>
<td>Nstring</td>
<td>Varchar(18)</td>
<td>Varchar(18)</td>
<td>Varchar2(18)</td>
</tr>
<tr>
<td>Smallint</td>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
<td>Number</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Datetime</td>
<td>Date</td>
</tr>
<tr>
<td>Timestamp With Local Time Zone</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Timestamp With Time Zone</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Urowid</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Varchar2</td>
<td>Nstring</td>
<td>Varchar</td>
<td>Varchar if precision is 255 or less. Text if precision is greater than 255.</td>
<td>Varchar2</td>
</tr>
</tbody>
</table>

**Datatype Conversion from Salesforce**

The following sections describe datatype conversions from Salesforce to other target types.
Salesforce to Microsoft SQL Server Datatypes

The following table shows the conversion of Salesforce datatypes to Microsoft SQL Server datatypes:

Table 10. Datatype Conversions: Salesforce to Microsoft SQL Server

<table>
<thead>
<tr>
<th>Salesforce Datatype</th>
<th>Description</th>
<th>Microsoft SQL Server Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnyType</td>
<td>Values can be any of the following types: string, picklist, boolean, int, double, percent, ID, date, dateTime, url, or email.</td>
<td>Ntext if precision &gt; 4000.</td>
<td>Ntext range is 1 to 1,073,741, 823 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar if precision &lt; 4000.</td>
<td>Nvarchar range is 1 to 4000 characters.</td>
</tr>
<tr>
<td>Base64</td>
<td>Base64-encoded arbitrary binary data.</td>
<td>Ntext if precision &gt; 4000.</td>
<td>Ntext range is 1 to 1,073,741, 823 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar if precision &lt; 4000.</td>
<td>Nvarchar range is 1 to 4000 characters.</td>
</tr>
<tr>
<td>Boolean</td>
<td>Boolean values: True or False.</td>
<td>Int</td>
<td>-2,147,483,648 to 2,147,483,647.</td>
</tr>
<tr>
<td>Currency</td>
<td>Currency values.</td>
<td>Decimal</td>
<td>Precision is 1 to 38. Scale is 0 to 38.</td>
</tr>
<tr>
<td>DataCategoryGroupReference</td>
<td>Types of category groups and category unique names.</td>
<td>Ntext if precision &gt; 4000.</td>
<td>Ntext range is 1 to 1,073,741, 823 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar if precision &lt; 4000.</td>
<td>Nvarchar range is 1 to 4000 characters.</td>
</tr>
<tr>
<td>Date</td>
<td>Date values.</td>
<td>Datetime</td>
<td>Jan 1, 1753 AD to Dec 31, 9999 AD. Precision is 23. Scale is 3. (precision to 3.33 milliseconds)</td>
</tr>
<tr>
<td>Datetime</td>
<td>Date and time values.</td>
<td>Datetime</td>
<td>Jan 1, 1753 AD to Dec 31, 9999 AD. Precision is 23. Scale is 3. (precision to 3.33 milliseconds)</td>
</tr>
<tr>
<td>Double</td>
<td>Double values.</td>
<td>Decimal</td>
<td>Precision is 1 to 38. Scale is 0 to 38.</td>
</tr>
<tr>
<td>Email</td>
<td>Email addresses.</td>
<td>Ntext if precision &gt; 4000.</td>
<td>Ntext range is 1 to 1,073,741, 823 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar if precision &lt; 4000.</td>
<td>Nvarchar range is 1 to 4000 characters.</td>
</tr>
<tr>
<td>ID</td>
<td>Primary key for a Salesforce object.</td>
<td>Ntext if precision &gt; 4000.</td>
<td>Ntext range is 1 to 1,073,741, 823 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar if precision &lt; 4000.</td>
<td>Nvarchar range is 1 to 4000 characters.</td>
</tr>
<tr>
<td>Salesforce Datatype</td>
<td>Description</td>
<td>Microsoft SQL Server Datatype</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>--------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Int</td>
<td>Integer values.</td>
<td>Int</td>
<td>-2,147,483,648 to 2,147,483,647.</td>
</tr>
<tr>
<td>Multipicklist</td>
<td>Multiple-selection picklists, which provide a set of enumerated values that you can select multiple values from.</td>
<td>Ntext if precision &gt; 4000. Nvarchar if precision &lt; 4000.</td>
<td>Ntext range is 1 to 1,073,741, 823 characters. Nvarchar range is 1 to 4000 characters.</td>
</tr>
<tr>
<td>Percent</td>
<td>Percentage values.</td>
<td>Decimal</td>
<td>Precision is 1 to 38. Scale is 0 to 38.</td>
</tr>
<tr>
<td>Picklist</td>
<td>Single-selection picklists, which provide a set of enumerated values that you can select one value from.</td>
<td>Ntext if precision &gt; 4000. Nvarchar if precision &lt; 4000.</td>
<td>Ntext range is 1 to 1,073,741, 823 characters. Nvarchar range is 1 to 4000 characters.</td>
</tr>
<tr>
<td>Reference</td>
<td>Cross-reference to another Salesforce object.</td>
<td>Ntext if precision &gt; 4000. Nvarchar if precision &lt; 4000.</td>
<td>Ntext range is 1 to 1,073,741, 823 characters. Nvarchar range is 1 to 4000 characters.</td>
</tr>
<tr>
<td>String</td>
<td>String values.</td>
<td>Ntext if precision &gt; 4000. Nvarchar if precision &lt; 4000.</td>
<td>Ntext range is 1 to 1,073,741, 823 characters. Nvarchar range is 1 to 4000 characters.</td>
</tr>
<tr>
<td>Textarea</td>
<td>String that appears as a multiple-line text field.</td>
<td>Ntext if precision &gt; 4000. Nvarchar if precision &lt; 4000.</td>
<td>Ntext range is 1 to 1,073,741, 823 characters. Nvarchar range is 1 to 4000 characters.</td>
</tr>
<tr>
<td>Time</td>
<td>Time values.</td>
<td>Datetime</td>
<td>Jan 1, 1753 AD to Dec 31, 9999 AD. Precision is 23. Scale is 3. (precision to 3.33 milliseconds)</td>
</tr>
<tr>
<td>Url</td>
<td>URL values.</td>
<td>Ntext if precision &gt; 4000. Nvarchar if precision &lt; 4000.</td>
<td>Ntext range is 1 to 1,073,741, 823 characters. Nvarchar range is 1 to 4000 characters.</td>
</tr>
</tbody>
</table>

**Salesforce to MySQL Datatypes**

You can load data from Salesforce fields of any datatype, except Text datatype with precision greater than 65535.
The following table shows the conversion of Salesforce datatypes to MySQL datatypes:

Table 11. Datatype Conversions: Salesforce to MySQL

<table>
<thead>
<tr>
<th>Salesforce Datatype</th>
<th>Description</th>
<th>MySQL Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnyType</td>
<td>Values can be any of the following types: string, picklist, boolean, int, double, percent, ID, date, dateTime, url, or email.</td>
<td>text if precision &gt; 255.</td>
<td>text range is 1 to 65535 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>varchar if precision &lt; 255.</td>
<td>varchar range is 1 to 255 characters.</td>
</tr>
<tr>
<td>Base64</td>
<td>Base64-encoded arbitrary binary data.</td>
<td>text if precision &gt; 255.</td>
<td>text range is 1 to 65535 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>varchar if precision &lt; 255.</td>
<td>varchar range is 1 to 255 characters.</td>
</tr>
<tr>
<td>Boolean</td>
<td>Boolean (true/false) values.</td>
<td>integer</td>
<td>Precision is 11. Scale is 0. Range is 0 or 1.</td>
</tr>
<tr>
<td>Currency</td>
<td>Currency values.</td>
<td>decimal</td>
<td>Precision is 1 to 65. Scale is 1 to 30.</td>
</tr>
<tr>
<td>Date</td>
<td>Date values.</td>
<td>date</td>
<td>January 01, 1000 00:00:00 AD to December 31, 9999 23:59:59 AD. Precision is 19. Scale is 0. (precision to second. Partial seconds are truncated.)</td>
</tr>
<tr>
<td>DataCategoryGroupReference</td>
<td>Types of category groups and category unique names.</td>
<td>text if precision &gt; 255.</td>
<td>text range is 1 to 65535 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>varchar if precision &lt; 255.</td>
<td>varchar range is 1 to 255 characters.</td>
</tr>
<tr>
<td>DateTime</td>
<td>Date and time values.</td>
<td>timestamp</td>
<td>January 01, 1000 00:00:00 AD to December 31, 9999 23:59:59 AD.</td>
</tr>
<tr>
<td>Double</td>
<td>Double values.</td>
<td>decimal</td>
<td>Precision is 1 to 65. Scale is 1 to 30.</td>
</tr>
<tr>
<td>Email</td>
<td>Email addresses.</td>
<td>text if precision &gt; 255.</td>
<td>text range is 1 to 65535 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>varchar if precision &lt; 255.</td>
<td>varchar range is 1 to 255 characters.</td>
</tr>
<tr>
<td>ID</td>
<td>Primary key for a Salesforce object.</td>
<td>text if precision &gt; 255.</td>
<td>text range is 1 to 65535 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>varchar if precision &lt; 255.</td>
<td>varchar range is 1 to 255 characters.</td>
</tr>
<tr>
<td>Int</td>
<td>Integer values.</td>
<td>integer</td>
<td>Precision is 11. Scale is 0. Range is -2147483648 to 2147483647.</td>
</tr>
<tr>
<td>Salesforce Datatype</td>
<td>Description</td>
<td>Salesforce Datatype</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Multipicklist</td>
<td>Multiple-selection picklists, which provide a set of enumerated values that you can select multiple values from.</td>
<td>MySQL Datatype</td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>text if precision &gt; 255.</td>
<td>text range is 1 to 65535 characters.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>varchar if precision &lt; 255.</td>
<td>varchar range is 1 to 255 characters.</td>
<td></td>
</tr>
<tr>
<td>Percent</td>
<td>Percentage values.</td>
<td>MySQL Datatype</td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>decimal</td>
<td>Precision is 1 to 65. Scale is 1 to 30.</td>
<td></td>
</tr>
<tr>
<td>Picklist</td>
<td>Single-selection picklists, which provide a set of enumerated values that you can select one value from.</td>
<td>MySQL Datatype</td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>text if precision &gt; 255.</td>
<td>text range is 1 to 65535 characters.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>varchar if precision &lt; 255.</td>
<td>varchar range is 1 to 255 characters.</td>
<td></td>
</tr>
<tr>
<td>Reference</td>
<td>Cross-reference to another Salesforce object.</td>
<td>MySQL Datatype</td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>text if precision &gt; 255.</td>
<td>text range is 1 to 65535 characters.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>varchar if precision &lt; 255.</td>
<td>varchar range is 1 to 255 characters.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>String values.</td>
<td>MySQL Datatype</td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>text if precision &gt; 255.</td>
<td>text range is 1 to 65535 characters.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>varchar if precision &lt; 255.</td>
<td>varchar range is 1 to 255 characters.</td>
<td></td>
</tr>
<tr>
<td>Textarea</td>
<td>String that appears as a multiple-line text field.</td>
<td>MySQL Datatype</td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>text if precision &gt; 255.</td>
<td>text range is 1 to 65535 characters.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>varchar if precision &lt; 255.</td>
<td>varchar range is 1 to 255 characters.</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>Time values.</td>
<td>MySQL Datatype</td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>datetime</td>
<td>January 01, 1000 00:00:00 AD to December 31, 9999 23:59:59 AD. Precision is to second.</td>
<td></td>
</tr>
<tr>
<td>Url</td>
<td>URL values.</td>
<td>MySQL Datatype</td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td>text if precision &gt; 255.</td>
<td>text range is 1 to 65535 characters.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>varchar if precision &lt; 255.</td>
<td>varchar range is 1 to 255 characters.</td>
<td></td>
</tr>
</tbody>
</table>

**Salesforce to Oracle Datatypes**

You can load data from Salesforce fields of any datatype, except binaries with precision greater than 2000.
The following table shows the conversion of Salesforce datatypes to Oracle datatypes:

Table 12. Datatype Conversions: Salesforce to Oracle

<table>
<thead>
<tr>
<th>Salesforce Datatype</th>
<th>Description</th>
<th>Oracle Datatype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AnyType</td>
<td>Values can be any of the following types: string, picklist, boolean, int, double, percent, ID, date, dateTime, url, or email.</td>
<td>Nclob if precision &gt; 2000.</td>
<td>Nclob range is up to 4 GB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar2 if precision &lt; 2000.</td>
<td>Nvarchar2 range is 1 to 4000 bytes.</td>
</tr>
<tr>
<td>Base64</td>
<td>Base64-encoded arbitrary binary data.</td>
<td>Nclob if precision &gt; 2000.</td>
<td>Nclob range is up to 4 GB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar2 if precision &lt; 2000.</td>
<td>Nvarchar2 range is 1 to 4000 bytes.</td>
</tr>
<tr>
<td>Boolean</td>
<td>Boolean (true/false) values.</td>
<td>Number</td>
<td>Precision is 10. Scale is 0.</td>
</tr>
<tr>
<td>Currency</td>
<td>Currency values.</td>
<td>Number(p,s) if scale equals 0.</td>
<td>Precision is 1 to 38.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number(p,s) if scale is greater than 0.</td>
<td>Precision is 1 to 38. Scale is 0 to 38.</td>
</tr>
<tr>
<td>Date</td>
<td>Date values.</td>
<td>Date</td>
<td>Jan 1, 4712 BC to Dec 31, 4712 AD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Precision is 19. Scale is 0.</td>
</tr>
<tr>
<td>DataCategoryGroupReferen-</td>
<td>Types of category groups and category unique names.</td>
<td>Nclob if precision &gt; 2000.</td>
<td>Nclob range is up to 4 GB.</td>
</tr>
<tr>
<td>ce</td>
<td></td>
<td>Nvarchar2 if precision &lt; 2000.</td>
<td>Nvarchar2 range is 1 to 4000 bytes.</td>
</tr>
<tr>
<td>DateTime</td>
<td>Date and time values.</td>
<td>Date</td>
<td>Jan 1, 4712 BC to Dec 31, 4712 AD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Precision is 19. Scale is 0.</td>
</tr>
<tr>
<td>Double</td>
<td>Double values.</td>
<td>Number(p,s) if scale equals 0.</td>
<td>Precision is 1 to 38.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number(p,s) if scale is greater than 0.</td>
<td>Precision is 1 to 38. Scale is 0 to 38.</td>
</tr>
<tr>
<td>Email</td>
<td>Email addresses.</td>
<td>Nclob if precision &gt; 2000.</td>
<td>Nclob range is up to 4 GB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar2 if precision &lt; 2000.</td>
<td>Nvarchar2 range is 1 to 4000 bytes.</td>
</tr>
<tr>
<td>ID</td>
<td>Primary key for a Salesforce object.</td>
<td>Nclob if precision &gt; 2000.</td>
<td>Nclob range is up to 4 GB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar2 if precision &lt; 2000.</td>
<td>Nvarchar2 range is 1 to 4000 bytes.</td>
</tr>
<tr>
<td>Int</td>
<td>Integer values.</td>
<td>Number</td>
<td>Precision is 10. Scale is 0.</td>
</tr>
<tr>
<td>Multipicklist</td>
<td>Multiple-selection picklists, which provide a set of enumerated values that you can select multiple values from.</td>
<td>Nclob if precision &gt; 2000.</td>
<td>Nclob range is up to 4 GB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar2 if precision &lt; 2000.</td>
<td>Nvarchar2 range is 1 to 4000 bytes.</td>
</tr>
<tr>
<td>Salesforce Datatype</td>
<td>Description</td>
<td>Oracle Datatype</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Percent</strong></td>
<td>Percentage values.</td>
<td>Number if scale equals 0.</td>
<td>Precision is 1 to 38.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number(p,s) if scale is greater than 0.</td>
<td>Precision is 1 to 38. Scale is 0 to 38.</td>
</tr>
<tr>
<td><strong>Picklist</strong></td>
<td>Single-selection picklists, which provide a set of enumerated values that you can select one value from.</td>
<td>Nclob if precision &gt; 2000.</td>
<td>Nclob range is up to 4 GB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar2 if precision &lt; 2000.</td>
<td>Nvarchar2 range is 1 to 4000 bytes.</td>
</tr>
<tr>
<td><strong>Reference</strong></td>
<td>Cross-reference to another Salesforce object.</td>
<td>Nclob if precision &gt; 2000.</td>
<td>Nclob range is up to 4 GB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar2 if precision &lt; 2000.</td>
<td>Nvarchar2 range is 1 to 4000 bytes.</td>
</tr>
<tr>
<td><strong>String</strong></td>
<td>String values.</td>
<td>Nclob if precision &gt; 2000.</td>
<td>Nclob range is up to 4 GB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar2 if precision &lt; 2000.</td>
<td>Nvarchar2 range is 1 to 4000 bytes.</td>
</tr>
<tr>
<td><strong>Textarea</strong></td>
<td>String that appears as a multiple-line text field.</td>
<td>Nclob if precision &gt; 2000.</td>
<td>Nclob range is up to 4 GB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar2 if precision &lt; 2000.</td>
<td>Nvarchar2 range is 1 to 4000 bytes.</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>Time values.</td>
<td>Date</td>
<td>Jan 1, 4712 BC to Dec 31, 4712 AD. Precision is 19. Scale is 0.</td>
</tr>
<tr>
<td><strong>Url</strong></td>
<td>URL values.</td>
<td>Nclob if precision &gt; 2000.</td>
<td>Nclob range is up to 4 GB.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nvarchar2 if precision &lt; 2000.</td>
<td>Nvarchar2 range is 1 to 4000 bytes.</td>
</tr>
</tbody>
</table>
Functions

This appendix includes the following topics:

- Function Overview, 195
- Function Quick Reference, 200
- ABORT, 207
- ABS, 208
- ADD_TO_DATE, 209
- AES_DECRYPT, 210
- AES_ENCRYPT, 211
- ASCII, 212
- CEIL, 213
- CHOOSE, 213
- CHR, 214
- CHRCODE, 215
- COMPRESS, 216
- CONCAT, 216
- CONVERT_BASE, 217
- COS, 218
- COSH, 218
- CRC32, 219
- CUME, 220
- DATECOMPARE, 221
- DATE_DIFF, 221
- DEC_BASE64, 223
- DECODE, 224
- DECOMPRESSION, 225
- ENC_BASE64, 226
- ERROR, 226
- EXP, 227
- FLOOR, 228
- FV, 228
- GET_DATE_PART, 229
• GREATEST, 231
• IIF, 231
• IN, 233
• INDEXOF, 234
• INITCAP, 235
• INSTR, 236
• IS_DATE, 238
• IS_NUMBER, 239
• IS_SPACES, 240
• ISNULL, 241
• LAST_DAY, 242
• LEAST, 243
• LENGTH, 243
• LN, 244
• LOG, 245
• LOWER, 245
• LPAD, 246
• LTRIM, 247
• MAKE_DATE_TIME, 248
• MD5, 249
• METAPHONE, 250
• MOD, 253
• NPER, 254
• PMT, 255
• POWER, 256
• PV, 257
• RAND, 258
• RATE, 258
• REG_EXTRACT, 259
• REG_MATCH, 261
• REG_REPLACE, 262
• REPLACECHR, 263
• REPLACESTR, 265
• REVERSE, 267
• ROUND (Dates), 268
• ROUND (Numbers), 270
• RPAD, 271
• RTRIM, 272
• SET_DATE_PART, 274
• SIGN, 275
• SIN, 276
Function Overview

You can use predefined functions in expressions to transform data before it is loaded into the target.

The transformation language provides the following function categories:

- Character
- Conversion
- Data Cleansing
- Date
- Encoding
- Financial
- Numerical
- Scientific
- Special
- String
- Test
- Variable

Character Functions

The transformation language provides the following character functions:

- ASCII
To evaluate character data, the character functions LOWER, UPPER, and INITCAP use the code page of the Secure Agent that runs the task.

Conversion Functions

The transformation language provides the following conversion functions:

- TO_BIGINT
- TO_CHAR(Date)
- TO_CHAR(Number)
- TO_DATE
- TO_DECIMAL
- TO_FLOAT
- TO_INTEGER

Data Cleansing Functions

The transformation language provides a group of functions to eliminate data errors. You can complete the following tasks with data cleansing functions:

- Test source values.
- Convert the datatype of an source value.
- Trim string values.
- Replace characters in a string.
- Encode strings.
- Match patterns in regular expressions.
The transformation language provides the following data cleansing functions:

- BETWEEN
- GREATEST
- IN
- INSTR
- IS_DATE
- IS_NUMBER
- IS_SPACES
- ISNULL
- LEAST
- LTRIM
- METAPHONE
- REG_EXTRACT
- REG_MATCH
- REG_REPLACE
- REPLACECHR
- REPLACESTR
- RTRIM
- SOUNDEX
- SUBSTR
- TO_BIGINT
- TO_CHAR
- TO_DATE
- TO_DECIMAL
- TO_FLOAT
- TO_INTEGER

### Date Functions

The transformation language provides a group of date functions to round, truncate, or compare dates, extract one part of a date, or perform arithmetic on a date.

You can pass any value with a date datatype to any of the date functions. However, if you want to pass a string to a date function, you must first use the TO_DATE function to convert it to a transformation Date/Time datatype.

The transformation language provides the following date functions:

- ADD_TO_DATE
- DATE_COMPARE
- DATE_DIFF
- GET_DATE_PART
- LAST_DAY
- MAKE_DATE_TIME
- ROUND
Several of the date functions include a format argument. You must specify one of the transformation language format strings for this argument. Date format strings are not internationalized.

The Date/Time transformation datatype does not support milliseconds. Therefore, if you pass a date with milliseconds, Informatica Cloud truncates the millisecond portion of the date.

**Encoding Functions**

The transformation language provides the following functions for data encryption, compression, encoding, and checksum:

- **AES_DECRYPT**
- **AES_ENCRYPT**
- **COMPRESS**
- **CRC32**
- **DEC_BASE64**
- **DECOMPRESS**
- **ENC_BASE64**
- **MD5**

**Financial Functions**

The transformation language provides the following financial functions:

- **FV**
- **NPER**
- **PMT**
- **PV**
- **RATE**

**Numeric Functions**

The transformation language provides the following numeric functions:

- **ABS**
- **CEIL**
- **CONV**
- **CUME**
- **EXP**
- **FLOOR**
- **LN**
- **LOG**
- **MOD**
• POWER
• RAND
• ROUND
• SIGN
• SQRT
• TRUNC

Scientific Functions

The transformation language provides the following scientific functions:

• COS
• COSH
• SIN
• SINH
• TAN
• TANH

Special Functions

The transformation language provides the following special functions:

• ABORT
• DECODE
• ERROR
• IIF
• LOOKUP

You can nest other functions within special functions.

String Functions

The transformation language provides the following string functions:

• CHOOSE
• INDEXOF
• REVERSE

Test Functions

The transformation language provides the following test functions:

• ISNULL
• IS_DATE
• IS_NUMBER
• IS_SPACES
# Function Quick Reference

The following table contains the syntax and a brief description of the functions that can be used in field expressions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Function Type</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORT</td>
<td>Special</td>
<td><code>ABORT( string )</code></td>
<td>Stops the session and issues a specified error message.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “ABORT” on page 207.</td>
</tr>
<tr>
<td>ABS</td>
<td>Numeric</td>
<td><code>ABS( numeric_value )</code></td>
<td>Returns the absolute value of a numeric value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “ABS” on page 208.</td>
</tr>
<tr>
<td>ADD_TO_DATE</td>
<td>Data Cleansing, Date</td>
<td><code>ADD_TO_DATE( date, format, amount )</code></td>
<td>Adds a specified amount to one part of a date/time value, and returns a date in the same format as the specified date. If you do not specify the year as YYYY, Informatica Cloud assumes the date is in the current century.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “ADD_TO_DATE” on page 209.</td>
</tr>
<tr>
<td>AES_DECRYPT</td>
<td>Encoding</td>
<td><code>AES_DECRYPT( value, key )</code></td>
<td>Returns encrypted data to string format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “AES_DECRYPT” on page 210.</td>
</tr>
<tr>
<td>AES_ENCRYPT</td>
<td>Encoding</td>
<td><code>AES_ENCRYPT( value, key )</code></td>
<td>Returns data in encrypted format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “AES_ENCRYPT” on page 211.</td>
</tr>
<tr>
<td>ASCII</td>
<td>Character</td>
<td><code>ASCII( string )</code></td>
<td>Returns the numeric ASCII value of the first character of the string passed to the function. This function is identical in behavior to the CHRCODE function. If you use the ASCII function in existing expressions, it will still work correctly. However, when you create new expressions, use the CHRCODE function instead of the ASCII function.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “ASCII” on page 212.</td>
</tr>
<tr>
<td>CEIL</td>
<td>Numeric</td>
<td><code>CEIL( numeric_value )</code></td>
<td>Returns the smallest integer greater than or equal to the specified numeric value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “CEIL” on page 213.</td>
</tr>
<tr>
<td>CHOOSE</td>
<td>String</td>
<td><code>CHOOSE( index, string1, [string2, ..., stringN] )</code></td>
<td>Chooses a string from a list of strings based on a given position.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “CHOOSE” on page 213.</td>
</tr>
<tr>
<td>CHR</td>
<td>Character</td>
<td><code>CHR( numeric_value )</code></td>
<td>Returns the ASCII character corresponding to the specified numeric value.</td>
</tr>
<tr>
<td>Function</td>
<td>Function Type</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>CHRCODE</td>
<td>Character</td>
<td>CHRCODE( string )</td>
<td>Returns the numeric ASCII value of the first character of the string passed to the function. This function is identical in behavior to the ASCII function. For more information, see “CHRCODE” on page 215.</td>
</tr>
<tr>
<td>COMPRESS</td>
<td>Encoding</td>
<td>COMPRESS( value )</td>
<td>Compresses data using the zlib compression algorithm. For more information, see “COMPRESS” on page 216.</td>
</tr>
<tr>
<td>CONCAT</td>
<td>Character</td>
<td>CONCAT( first_string, second_string )</td>
<td>Concatenates two strings. For more information, see “CONCAT” on page 216.</td>
</tr>
<tr>
<td>CONVERT_BASE</td>
<td>Numeric</td>
<td>CONVERT_BASE( value, source_base, dest_base )</td>
<td>Converts a number from one base value to another base value. For more information, see “CONVERT_BASE” on page 217.</td>
</tr>
<tr>
<td>COS</td>
<td>Scientific</td>
<td>COS( numeric_value )</td>
<td>Returns the cosine of a numeric value (expressed in radians). For more information, see “COS” on page 218.</td>
</tr>
<tr>
<td>COSH</td>
<td>Scientific</td>
<td>COSH( numeric_value )</td>
<td>Returns the hyperbolic cosine of a numeric value (expressed in radians). For more information, see “COSH” on page 218.</td>
</tr>
<tr>
<td>CRC32</td>
<td>Encoding</td>
<td>CRC32( value )</td>
<td>Returns a 32-bit Cyclic Redundancy Check (CRC32) value. For more information, see “CRC32” on page 219.</td>
</tr>
<tr>
<td>CUME</td>
<td>Numeric</td>
<td>CUME( numeric_value [, filter_condition] )</td>
<td>Returns a running total. For more information, see “CUME” on page 220.</td>
</tr>
<tr>
<td>DATE_COMPARE</td>
<td>Data Cleansing, Date</td>
<td>DATE_COMPARE( date1, date2 )</td>
<td>Returns a value indicating the earlier of two dates. For more information, see “DATE_COMPARE” on page 221.</td>
</tr>
<tr>
<td>DATE_DIFF</td>
<td>Data Cleansing, Date</td>
<td>DATE_DIFF( date1, date2, format )</td>
<td>Returns the length of time between two dates, measured in the specified increment (years, months, days, hours, minutes, or seconds). For more information, see “DATE_DIFF” on page 221.</td>
</tr>
<tr>
<td>DEC_BASE64</td>
<td>Encoding</td>
<td>DEC_BASE64( value )</td>
<td>Decodes the value and returns a string with the binary data representation of the data.</td>
</tr>
<tr>
<td>Function</td>
<td>Function Type</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DECODE</td>
<td>Special</td>
<td>DECODE( value, first_search, first_result [, second_search, second_result ]...[, default ] )</td>
<td>Searches a column for the specified value. For more information, see “DECODE” on page 224.</td>
</tr>
<tr>
<td>DECOMPRESS</td>
<td>Encoding</td>
<td>DECOMPRESS( value, precision )</td>
<td>Decompresses data using the zlib compression algorithm. For more information, see “DECOMPRESS” on page 225.</td>
</tr>
<tr>
<td>ENC_BASE64</td>
<td>Encoding</td>
<td>ENC_BASE64( value )</td>
<td>Encodes data by converting binary data to string data using Multipurpose Internet Mail Extensions (MIME) encoding. For more information, see “ENC_BASE64” on page 226.</td>
</tr>
<tr>
<td>ERROR</td>
<td>Special</td>
<td>ERROR( string )</td>
<td>Causes the Informatica Cloud to skip a row. It writes the row into the error log file with the specified error message. For more information, see “ERROR” on page 226.</td>
</tr>
<tr>
<td>EXP</td>
<td>Numeric</td>
<td>EXP( exponent )</td>
<td>Returns e raised to the specified power (exponent), where e=2.71828183. For more information, see “EXP” on page 227.</td>
</tr>
<tr>
<td>FLOOR</td>
<td>Numeric</td>
<td>FLOOR( numeric_value )</td>
<td>Returns the largest integer less than or equal to the specified numeric value. For more information, see “FLOOR” on page 228.</td>
</tr>
<tr>
<td>FV</td>
<td>Financial</td>
<td>FV( rate, terms, payment [, present value, type] )</td>
<td>Returns the future value of an investment, where you make periodic, constant payments and the investment earns a constant interest rate. For more information, see “FV” on page 228.</td>
</tr>
<tr>
<td>GET_DATE_PART</td>
<td>Date, Data Cleansing</td>
<td>GET_DATE_PART( date, format )</td>
<td>Returns the specified part of a date as an integer value, based on the default date format of MM/DD/YYYY HH24:MI:SS. For more information, see “GET_DATE_PART” on page 229.</td>
</tr>
<tr>
<td>GREATEST</td>
<td>Data Cleansing</td>
<td>GREATEST( value1, [value2, ..., valueN], CaseFlag )</td>
<td>Returns the greatest value from a list of input values. For more information, see “GREATEST” on page 231.</td>
</tr>
<tr>
<td>IIF</td>
<td>Special</td>
<td>IIF( condition, value2 [, value2 ] )</td>
<td>Returns one of two values you specify, based on the results of a condition.</td>
</tr>
<tr>
<td>Function</td>
<td>Function Type</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>IN</strong></td>
<td>Data Cleansing</td>
<td><code>IN( valueToSearch, value1, [value2, ..., valueN], CaseFlag )</code></td>
<td>Matches input data to a list of values. For more information, see “IN” on page 233.</td>
</tr>
<tr>
<td><strong>INDEXOF</strong></td>
<td>String</td>
<td><code>INDEXOF( valueToSearch, string1, [string2, ..., stringN], CaseFlag )</code></td>
<td>Finds the index of a string among a list of strings. For more information, see “INDEXOF” on page 234.</td>
</tr>
<tr>
<td><strong>INITCAP</strong></td>
<td>Character</td>
<td><code>INITCAP( string )</code></td>
<td>Capitalizes the first letter in each word of a string and converts all other letters to lowercase. For more information, see “INITCAP” on page 235.</td>
</tr>
<tr>
<td><strong>INSTR</strong></td>
<td>Character, Data Cleansing</td>
<td><code>INSTR( string, search_value [, start [, occurrence ] ] )</code></td>
<td>Returns the position of a character set in a string, counting from left to right. For more information, see “INSTR” on page 236.</td>
</tr>
<tr>
<td><strong>IS_DATE</strong></td>
<td>Data Cleansing, Test</td>
<td><code>IS_DATE( value )</code></td>
<td>Returns whether a value is a valid date. For more information, see “IS_DATE” on page 238.</td>
</tr>
<tr>
<td><strong>IS_NUMBER</strong></td>
<td>Data Cleansing, Test</td>
<td><code>IS_NUMBER( value )</code></td>
<td>Returns whether a string is a valid number. For more information, see “IS_NUMBER” on page 239.</td>
</tr>
<tr>
<td><strong>IS_SPACES</strong></td>
<td>Data Cleansing, Test</td>
<td><code>IS_SPACES( value )</code></td>
<td>Returns whether a value consists entirely of spaces. For more information, see “IS_SPACES” on page 240.</td>
</tr>
<tr>
<td><strong>ISNULL</strong></td>
<td>Data Cleansing, Test</td>
<td><code>ISNULL( value )</code></td>
<td>Returns whether a value is NULL. For more information, see “ISNULL” on page 241.</td>
</tr>
<tr>
<td><strong>LAST_DAY</strong></td>
<td>Data Cleansing, Date</td>
<td><code>LAST_DAY( date )</code></td>
<td>Returns the date of the last day of the month for each date in a column. For more information, see “LAST_DAY” on page 242.</td>
</tr>
<tr>
<td><strong>LEAST</strong></td>
<td>Data Cleansing</td>
<td><code>LEAST( value1, [value2, ..., valueN], CaseFlag )</code></td>
<td>Returns the smallest value from a list of input values. For more information, see “LEAST” on page 243.</td>
</tr>
<tr>
<td><strong>LENGTH</strong></td>
<td>Character</td>
<td><code>LENGTH( string )</code></td>
<td>Returns the number of characters in a string, including trailing blanks. For more information, see “LENGTH” on page 243.</td>
</tr>
<tr>
<td><strong>LN</strong></td>
<td>Numeric</td>
<td><code>LN( numeric_value )</code></td>
<td>Returns the natural logarithm of a numeric value. For more information, see “LN” on page 244.</td>
</tr>
<tr>
<td><strong>LOG</strong></td>
<td>Numeric</td>
<td><code>LOG( base, exponent )</code></td>
<td>Returns the logarithm of a numeric value.</td>
</tr>
<tr>
<td>Function</td>
<td>Function Type</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------</td>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LOWER</td>
<td>Character</td>
<td>LOWER( string )</td>
<td>Converts uppercase string characters to lowercase. For more information, see “LOWER” on page 245.</td>
</tr>
<tr>
<td>LPAD</td>
<td>Character</td>
<td>LPAD( first_string, length [, second_string ] )</td>
<td>Adds a set of blanks or characters to the beginning of a string to set a string to a specified length. For more information, see “LPAD” on page 246.</td>
</tr>
<tr>
<td>LTRIM</td>
<td>Character, Data Cleansing</td>
<td>LTRIM( string [, trim_set ] )</td>
<td>Removes blanks or characters from the beginning of a string. For more information, see “LTRIM” on page 247.</td>
</tr>
<tr>
<td>MAKE_DATE_TIME</td>
<td>Data Cleansing, Date</td>
<td>MAKE_DATE_TIME( year, month, day, hour, minute, second )</td>
<td>Returns the date and time based on the input values. For more information, see “MAKE_DATE_TIME” on page 248.</td>
</tr>
<tr>
<td>MD5</td>
<td>Encoding</td>
<td>MD5( value )</td>
<td>Calculates the checksum of the input value. The function uses Message-Digest algorithm 5 (MD5). For more information, see “MD5” on page 249.</td>
</tr>
<tr>
<td>METAPHONE</td>
<td>Data Cleansing</td>
<td>METAPHONE( string [, length ] )</td>
<td>Encodes characters of the English language alphabet (A-Z). For more information, see “METAPHONE” on page 250.</td>
</tr>
<tr>
<td>MOD</td>
<td>Numeric</td>
<td>MOD( numeric_value, divisor )</td>
<td>Returns the remainder of a division calculation. For more information, see “MOD” on page 253.</td>
</tr>
<tr>
<td>NPER</td>
<td>Financial</td>
<td>NPER( rate, present value, payment [, future value, type] )</td>
<td>Returns the number of periods for an investment based on a constant interest rate and periodic, constant payments. For more information, see “NPER” on page 254.</td>
</tr>
<tr>
<td>PMT</td>
<td>Financial</td>
<td>PMT( Rate, terms, present value [, future value, type] )</td>
<td>Returns the payment for a loan based on constant payments and a constant interest rate. For more information, see “PMT” on page 255.</td>
</tr>
<tr>
<td>POWER</td>
<td>Numeric</td>
<td>POWER( base, exponent )</td>
<td>Returns a value raised to the specified exponent. For more information, see “POWER” on page 256.</td>
</tr>
<tr>
<td>PV</td>
<td>Financial</td>
<td>PV( Rate, terms, payment [, future value, type] )</td>
<td>Returns the present value of an investment. For more information, see “PV” on page 257.</td>
</tr>
<tr>
<td>RAND</td>
<td>Numeric</td>
<td>RAND( seed )</td>
<td>Returns a random number between 0 and 1.</td>
</tr>
<tr>
<td>Function</td>
<td>Function Type</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------</td>
<td>------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RATE</td>
<td>Financial</td>
<td>RATE( terms, payment, present value [ , future value, type ] )</td>
<td>Returns the interest rate earned per period by a security.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “RATE” on page 258.</td>
</tr>
<tr>
<td>REG_EXTRACT</td>
<td>Data Cleansing</td>
<td>REG_EXTRACT( subject, pattern, subPatternNum )</td>
<td>Extracts subpatterns of a regular expression within an input value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “REG_EXTRACT” on page 259.</td>
</tr>
<tr>
<td>REG_MATCH</td>
<td>Data Cleansing</td>
<td>REG_MATCH( subject, pattern )</td>
<td>Returns whether a value matches a regular expression pattern.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “REG_MATCH” on page 261.</td>
</tr>
<tr>
<td>REG_REPLACE</td>
<td>Data Cleansing</td>
<td>REG_REPLACE( subject, pattern, replace, numReplacements )</td>
<td>Replaces characters in a string with a another character pattern.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “REG_REPLACE” on page 262.</td>
</tr>
<tr>
<td>REPLACECHR</td>
<td>Character</td>
<td>REPLACECHR( CaseFlag, InputString, OldCharSet, NewChar )</td>
<td>Replaces characters in a string with a single character or no character.</td>
</tr>
<tr>
<td></td>
<td>Data Cleansing</td>
<td></td>
<td>For more information, see “REPLACECHR” on page 263.</td>
</tr>
<tr>
<td>REPLACESTR</td>
<td>Character</td>
<td>REPLACESTR ( InputString, OldString1, [OldString2, ... OldStringN] ) NewString</td>
<td>Replaces characters in a string with a single character, multiple characters, or no character.</td>
</tr>
<tr>
<td></td>
<td>Data Cleansing</td>
<td></td>
<td>For more information, see “REPLACESTR” on page 265.</td>
</tr>
<tr>
<td>REVERSE</td>
<td>String</td>
<td>REVERSE( string )</td>
<td>Reverses the input string.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “REVERSE” on page 267.</td>
</tr>
<tr>
<td>ROUND</td>
<td>Data Cleansing, Date, Numeric</td>
<td>ROUND( date [ , format ] ) or ROUND( numeric_value [ , precision ] )</td>
<td>For data cleansing, rounds one part of a date. For numeric values, rounds numbers to a specified digit.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “ROUND (Dates)” on page 268 or “ROUND (Numbers)” on page 270.</td>
</tr>
<tr>
<td>RPAD</td>
<td>Character</td>
<td>RPAD( first_string, length [ , second_string ] )</td>
<td>Converts a string to a specified length by adding blanks or characters to the end of the string.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “RPAD” on page 271.</td>
</tr>
<tr>
<td>RTRIM</td>
<td>Character, Data Cleansing</td>
<td>RTRIM( string [ , trim_set ] )</td>
<td>Removes blanks or characters from the end of a string.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “RTRIM” on page 272.</td>
</tr>
<tr>
<td>Function</td>
<td>Function Type</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SET_DATE_PART</td>
<td>Data Cleansing, Date</td>
<td>SET_DATE_PART( date, format, value )</td>
<td>Sets one part of a date/time value to a specified value. For more information, see “SET_DATE_PART” on page 274.</td>
</tr>
<tr>
<td>SIGN</td>
<td>Numeric</td>
<td>SIGN( numeric_value )</td>
<td>Indicates whether a numeric value is positive, negative, or 0. For more information, see “SIGN” on page 275.</td>
</tr>
<tr>
<td>SIN</td>
<td>Scientific</td>
<td>SIN( numeric_value )</td>
<td>Returns the sin of a numeric value expressed in radians. For more information, see “SIN” on page 276.</td>
</tr>
<tr>
<td>SINH</td>
<td>Scientific</td>
<td>SINH( numeric_value )</td>
<td>Returns the hyperbolic sin of a numeric value expressed in radians. For more information, see “SINH” on page 277.</td>
</tr>
<tr>
<td>SOUNDEX</td>
<td>Data Cleansing</td>
<td>SOUNDEX( string )</td>
<td>Encodes a string into a four-character string. For more information, see “SOUNDEX” on page 277.</td>
</tr>
<tr>
<td>SQRT</td>
<td>Numeric</td>
<td>SQRT( numeric_value )</td>
<td>Returns the square root of a positive numeric value. For more information, see “SQRT” on page 279.</td>
</tr>
<tr>
<td>SUBSTR</td>
<td>Character, Data Cleansing</td>
<td>SUBSTR( string, start, length )</td>
<td>Returns a portion of a string. For more information, see “SUBSTR” on page 279.</td>
</tr>
<tr>
<td>SYSTIMESTAMP</td>
<td>Date</td>
<td>SYSTIMESTAMP( format )</td>
<td>Returns the current date and time with precision to the nanosecond of the system that hosts the Secure Agent that starts the task. For more information, see “SYSTIMESTAMP” on page 281.</td>
</tr>
<tr>
<td>TAN</td>
<td>Scientific</td>
<td>TAN( numeric_value )</td>
<td>Returns the tangent of a numeric value expressed in radians. For more information, see “TAN” on page 282.</td>
</tr>
<tr>
<td>TANH</td>
<td>Scientific</td>
<td>TANH( numeric_value )</td>
<td>Returns the hyperbolic tangent of a numeric value expressed in radians. For more information, see “TANH” on page 282.</td>
</tr>
<tr>
<td>TO_BIGINT</td>
<td>Conversion, Data Cleansing</td>
<td>TO_BIGINT( value, flag )</td>
<td>Converts a string or numeric value to a bigint value. For more information, see “TO_BIGINT” on page 283.</td>
</tr>
<tr>
<td>TO_CHAR</td>
<td>Conversion, Data Cleansing</td>
<td>TO_CHAR( date, format, or )</td>
<td>Converts dates or numeric values to text strings.</td>
</tr>
<tr>
<td>Function</td>
<td>Function Type</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------</td>
<td>---------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TO_CHAR</td>
<td></td>
<td>TO_CHAR( numeric_value )</td>
<td>For more information, see “TO_CHAR (Dates)” on page 284 or “TO_CHAR (Numbers)” on page 287.</td>
</tr>
<tr>
<td>TO_DATE</td>
<td>Conversion, Data Cleansing</td>
<td>TO_DATE( string [, format] )</td>
<td>Converts a character string to a date datatype in the same format as the character string. For conversion, you must specify the date format if the string is not in the mm/dd/yyyy hh:mm:ss format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see “TO_DATE” on page 288. For more information about date formats, see Appendix E, “Dates” on page 302.</td>
</tr>
<tr>
<td>TO_DECIMAL</td>
<td>Conversion, Data Cleansing</td>
<td>TO_DECIMAL( value [, scale] )</td>
<td>Converts any value (except binary) to a decimal. For more information, see “TO_DECIMAL” on page 290.</td>
</tr>
<tr>
<td>TO_FLOAT</td>
<td>Conversion, Data Cleansing</td>
<td>TO_FLOAT( value )</td>
<td>Converts any value (except binary) to a double-precision floating point number (the Double datatype). For more information, see “TO_FLOAT” on page 290.</td>
</tr>
<tr>
<td>TO_INTEGER</td>
<td>Conversion, Data Cleansing</td>
<td>TO_INTEGER( value )</td>
<td>Converts any value (except binary) to an integer by rounding the decimal portion of a value. For more information, see “TO_INTEGER” on page 291.</td>
</tr>
<tr>
<td>TRUNC</td>
<td>Data Cleansing, Date, Numeric</td>
<td>TRUNC( date [, format] ) or TRUNC( numeric_value [, precision])</td>
<td>Truncates dates to a specific year, month, day, hour, or minute. Truncates numeric values to a specific digit. For more information, see “TRUNC (Dates)” on page 292 or “TRUNC (Numbers)” on page 294.</td>
</tr>
<tr>
<td>UPPER</td>
<td>Character</td>
<td>UPPER( string )</td>
<td>Converts lowercase string characters to uppercase. For more information, see “UPPER” on page 295.</td>
</tr>
</tbody>
</table>

**ABORT**

Stops the task and issues an error message to the activity log. When Informatica Cloud encounters an ABORT function, it stops transforming data at that row.

Use ABORT to validate data. Generally, you use ABORT within an IIF or DECODE function to set rules for aborting a task. You might use ABORT to avoid writing null values to the target.
Syntax

**ABORT( string )**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>String. The message you want to display in the activity log when the session stops. The string can be any length. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

NULL.

### ABS

Returns the absolute value of a numeric value.

**Syntax**

**ABS( numeric_value )**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. Returns the absolute value of a number. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

**Return Value**

Positive numeric value. ABS returns the same datatype as the numeric value passed as an argument. If you pass a Double, it returns a Double. Likewise, if you pass an Integer, it returns an Integer.

NULL if you pass a null value to the function.

**Example**

The following expression returns the difference between two numbers as a positive value, regardless of which number is larger:

\[
\text{ABS( PRICE - COST )}
\]

<table>
<thead>
<tr>
<th>PRICE</th>
<th>COST</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>150</td>
<td>100</td>
</tr>
<tr>
<td>52</td>
<td>48</td>
<td>4</td>
</tr>
<tr>
<td>169.95</td>
<td>69.95</td>
<td>100</td>
</tr>
<tr>
<td>59.95</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>70</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>430</td>
<td>330</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>200</td>
<td>100</td>
</tr>
</tbody>
</table>
**ADD_TO_DATE**

Adds a specified amount to one part of a datetime value, and returns a date in the same format as the date you pass to the function. ADD_TO_DATE accepts positive and negative integer values. Use ADD_TO_DATE to change the following parts of a date:

- **Year.** Enter a positive or negative integer in the *amount* argument. Use any of the year format strings: Y, YY, YYY, or YYYY. For example, the expression `ADD_TO_DATE( SHIP_DATE, 'YY', 10 )` adds 10 years to all dates in the SHIP_DATE column.

- **Month.** Enter a positive or negative integer in the *amount* argument. Use any of the month format strings: MM, MON, MONTH. For example, the expression `ADD_TO_DATE( SHIP_DATE, 'MONTH', -10 )` subtracts 10 months from each date in the SHIP_DATE column.

- **Day.** Enter a positive or negative integer in the *amount* argument. Use any of the day format strings: D, DD, DDD, DY, and DAY. For example, the expression `ADD_TO_DATE( SHIP_DATE, 'DD', 10 )` adds 10 days to each date in the SHIP_DATE column.

- **Hour.** Enter a positive or negative integer in the *amount* argument. Use any of the hour format strings: HH, HH12, HH24. For example, the expression `ADD_TO_DATE( SHIP_DATE, 'HH', 14 )` adds 14 hours to each date in the SHIP_DATE column.

- **Minute.** Enter a positive or negative integer in the *amount* argument. Use the MI format string to set the minute. For example, the expression `ADD_TO_DATE( SHIP_DATE, 'MI', 25 )` adds 25 minutes to each date in the SHIP_DATE column.

- **Seconds.** Enter a positive or negative integer in the *amount* argument. Use the SS format string to set the second. For example, the following expression adds 59 seconds to each date in the SHIP_DATE column:

  `ADD_TO_DATE( SHIP_DATE, 'SS', 59 )`

### Syntax

```
ADD_TO_DATE( date, format, amount )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>Required</td>
<td>Date/Time datatype. Passes the values you want to change. You can enter any valid expression.</td>
</tr>
<tr>
<td>format</td>
<td>Required</td>
<td>A format string specifying the portion of the date value you want to change. Enclose the format string within single quotation marks, for example, 'mm'. The format string is not case sensitive.</td>
</tr>
<tr>
<td>amount</td>
<td>Required</td>
<td>An integer value specifying the amount of years, months, days, hours, and so on by which you want to change the date value. You can enter any valid expression that evaluates to an integer.</td>
</tr>
</tbody>
</table>

### Return Value

Date in the same format as the date you pass to this function.

NULL if a null value is passed as an argument to the function.

### Example

The following expressions all add one month to each date in the DATE_SHIPPED column. If you pass a value that creates a day that does not exist in a particular month, Informatica Cloud returns the last day of the month. For example, if you add one month to Jan 31 1998, Informatica Cloud returns Feb 28 1998.
ADD_TO_DATE recognizes leap years and adds one month to Jan 29 2000:

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 12 1998 12:00:30AM</td>
<td>Feb 12 1998 12:00:30AM</td>
</tr>
<tr>
<td>Jan 29 2000 5:32:12AM</td>
<td>Feb 29 2000 5:32:12AM  (Leap Year)</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions subtract 10 days from each date in the DATE_SHIPPED column:

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 1 1997 12:00:30AM</td>
<td>Dec 22 1996 12:00AM</td>
</tr>
<tr>
<td>Mar 9 1996 5:32:12AM</td>
<td>Feb 29 1996 5:32:12AM  (Leap Year)</td>
</tr>
<tr>
<td>Oct 9 1997 2:30:12PM</td>
<td>Sep 30 1997 2:30:12PM</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions subtract 15 hours from each date in the DATE_SHIPPED column:

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 1 1997 12:00:30AM</td>
<td>Dec 31 1996 9:00:30AM</td>
</tr>
<tr>
<td>Oct 9 1997 2:30:12PM</td>
<td>Oct 8 1997 11:30:12PM</td>
</tr>
<tr>
<td>Mar 1 1996 5:32:12AM</td>
<td>Feb 29 1996 2:32:12AM  (Leap Year)</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

**Working with Dates**

Use the following tips when working with ADD_TO_DATE:

- You can add or subtract any part of the date by specifying a format string and making the amount argument a positive or negative integer.
- If you pass a value that creates a day that does not exist in a particular month, Informatica Cloud returns the last day of the month. For example, if you add one month to Jan 31 1998, Informatica Cloud returns Feb 28 1998.
- You can nest TRUNC and ROUND to manipulate dates.
- You can nest TO_DATE to convert strings to dates.
- ADD_TO_DATE changes only one portion of the date, which you specify. If you modify a date so that it changes from standard to daylight savings time, you need to change the hour portion of the date.

**AES_DECRYPT**

Returns encrypted data to string format. Informatica Cloud uses Advanced Encryption Standard (AES) algorithm with 128-bit encoding. The AES algorithm is a FIPS-approved cryptographic algorithm.
**Syntax**

```plaintext
AES_DECRYPT ( value, key )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>Binary datatype. Value you want to decrypt.</td>
</tr>
<tr>
<td>key</td>
<td>Required</td>
<td>String datatype. Precision of 16 characters or fewer. Use the same key to decrypt a value that you used to encrypt it.</td>
</tr>
</tbody>
</table>

**Return Value**

Decrypted binary value.

NULL if the input value is a null value.

**Example**

The following example returns decrypted social security numbers. In this example, Informatica Cloud derives the key from the first three numbers of the social security number using the SUBSTR function:

```plaintext
AES_DECRYPT( SSN_ENCRYPT, SUBSTR( SSN,1,3 ))
```

<table>
<thead>
<tr>
<th>SSN_ENCRYPT</th>
<th>DECRYPTED VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>07FB9459268</td>
<td>832-17-1672</td>
</tr>
<tr>
<td>9153ACAB89D5</td>
<td>832-46-7552</td>
</tr>
<tr>
<td>AF6B5E439F74B3</td>
<td>832-53-6194</td>
</tr>
<tr>
<td>92D6A5D91E75D03B94D</td>
<td>832-81-9528</td>
</tr>
</tbody>
</table>

**AES_ENCRYPT**

Returns data in encrypted format. Informatica Cloud uses Advanced Encryption Standard (AES) algorithm with 128-bit encoding. The AES algorithm is a FIPS-approved cryptographic algorithm.

Use this function to prevent sensitive data from being visible to everyone. For example, to store social security numbers in a data warehouse, use the AES_ENCRYPT function to encrypt the social security numbers to maintain confidentiality.

**Syntax**

```plaintext
AES_ENCRYPT ( value, key )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>String datatype. Value you want to encrypt.</td>
</tr>
<tr>
<td>key</td>
<td>Required</td>
<td>String datatype. Precision of 16 characters or fewer.</td>
</tr>
</tbody>
</table>

**Return Value**

Encrypted binary value.

NULL if the input is a null value.
Example

The following example returns encrypted values for social security numbers. In this example, Informatica Cloud derives the key from the first three numbers of the social security number using the SUBSTR function:

```
AES_ENCRYPT( SSN, SUBSTR( SSN, 1, 3 ) )
```

<table>
<thead>
<tr>
<th>SSN</th>
<th>ENCRYPTED VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>832-17-1672</td>
<td>07FB945926849D2B1641E708C85E439D</td>
</tr>
<tr>
<td>832-92-4731</td>
<td>9153ACAB89D65A4B81AD2ABF151B099D</td>
</tr>
<tr>
<td>832-46-7552</td>
<td>AF685E4E39F974B3F3FB022320CC60E</td>
</tr>
<tr>
<td>832-53-6194</td>
<td>992D6A5D91E7F59D03B940A4B1CBBCE</td>
</tr>
<tr>
<td>832-81-9528</td>
<td>992D6A5D91E7F59D03B940A4B1CBBCE</td>
</tr>
</tbody>
</table>

Tip

If the target does not support binary data, use AES_ENCRYPT with the ENC_BASE64 function to store the data in a format compatible with the database.

ASCII

The ASCII function returns the numeric ASCII value of the first character of the string passed to the function.

You can pass a string of any size to ASCII, but it evaluates only the first character in the string. Before you pass any string value to ASCII, you can parse out the specific character you want to convert to an ASCII value. For example, you might use RTRIM or another string-manipulation function. If you pass a numeric value, ASCII converts it to a character string and returns the ASCII value of the first character in the string.

This function is identical in behavior to the CHRCODE function. If you use ASCII in existing expressions, they will still work correctly. However, when you create new expressions, use the CHRCODE function instead of the ASCII function.

Syntax

```
ASCII ( string )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>Character string. Passes the value you want to return as an ASCII value. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

Integer. The ASCII value of the first character in the string.

NULL if a value passed to the function is NULL.

Example

The following expression returns the ASCII value for the first character of each value in the ITEMS column:

```
ASCII( ITEMS )
```

<table>
<thead>
<tr>
<th>ITEMS</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flashlight</td>
<td>70</td>
</tr>
<tr>
<td>Compass</td>
<td>67</td>
</tr>
<tr>
<td>Safety Knife</td>
<td>83</td>
</tr>
<tr>
<td>Depth/Pressure Gauge</td>
<td>68</td>
</tr>
<tr>
<td>Regulator System</td>
<td>82</td>
</tr>
</tbody>
</table>
CEIL

Returns the smallest integer greater than or equal to the numeric value passed to this function. For example, if you pass 3.14 to CEIL, the function returns 4. If you pass 3.98 to CEIL, the function returns 4. Likewise, if you pass -3.17 to CEIL, the function returns -3.

Syntax

CEIL( numeric_value )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

Integer if you pass a numeric value with declared precision between 0 and 28.
Double value if you pass a numeric value with declared precision greater than 28.
NULL if a value passed to the function is NULL.

Example

The following expression returns the price rounded to the next integer:

CEIL( PRICE )

<table>
<thead>
<tr>
<th>PRICE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>39.79</td>
<td>40</td>
</tr>
<tr>
<td>125.12</td>
<td>126</td>
</tr>
<tr>
<td>74.24</td>
<td>75</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>-100.99</td>
<td>-100</td>
</tr>
</tbody>
</table>

Tip

You can perform arithmetic on the values passed to CEIL before CEIL returns the next integer value. For example, if you wanted to multiply a numeric value by 10 before you calculated the smallest integer less than the modified value, you might write the function as follows:

CEIL( PRICE * 10 )

CHOOSE

Chooses a string from a list of strings based on a given position. You specify the position and the value. If the value matches the position, Informatica Cloud returns the value.
**Syntax**

\[
\text{CHOOSE}( \text{index}, \text{string}_1, [\text{string}_2, \ldots, \text{string}_N] )
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>Required</td>
<td>Numeric datatype. Enter a number based on the position of the value you want to match.</td>
</tr>
<tr>
<td>string</td>
<td>Required</td>
<td>Any character value.</td>
</tr>
</tbody>
</table>

**Return Value**

The string that matches the position of the index value.

NULL if no string matches the position of the index value.

**Example**

The following expression returns the string 'flashlight' based on an index value of 2:

\[
\text{CHOOSE}( 2, 'knife', 'flashlight', 'diving hood' )
\]

The following expression returns NULL based on an index value of 4:

\[
\text{CHOOSE}( 4, 'knife', 'flashlight', 'diving hood' )
\]

**CHR**

CHR returns the ASCII character corresponding to the numeric value you pass to this function. ASCII values fall in the range 0 to 255. You can pass any integer to CHR, but only ASCII codes 32 to 126 are printable characters.

**Syntax**

\[
\text{CHR}( \text{numeric}_\text{value} )
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. The value you want to return as an ASCII character. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

**Return Value**

ASCII character. A string containing one character.

NULL if a value passed to the function is NULL.

**Example**

The following expression returns the ASCII character for each numeric value in the ITEM_ID column:

\[
\text{CHR}( \text{ITEM}_{\text{ID}} )
\]

<table>
<thead>
<tr>
<th>ITEM_ID</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>A</td>
</tr>
<tr>
<td>122</td>
<td>z</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>88</td>
<td>X</td>
</tr>
<tr>
<td>100</td>
<td>d</td>
</tr>
</tbody>
</table>
Use the CHR function to concatenate a single quotation mark onto a string. The single quotation mark is the only character that you cannot use inside a string literal. Consider the following example:

'Joan' || CHR(39) || 's car'

The return value is:

Joan's car

CHRCODE

CHRCODE returns the numeric ASCII value of the first character of the string passed to the function. ASCII values fall in the range 0 to 255.

Normally, before you pass any string value to CHRCODE, you parse out the specific character you want to convert to an ASCII value. For example, you might use RTRIM or another string-manipulation function. If you pass a numeric value, CHRCODE converts it to a character string and returns the ASCII value of the first character in the string.

This function is identical in behavior to the ASCII function. If you currently use ASCII in expressions, it will still work correctly. However, when you create new expressions, use the CHRCODE function instead of the ASCII function.

Syntax

CHRCODE( string )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>Character string. Passes the values you want to return as ASCII values. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

Integer. The ASCII value of the first character in the string.

NULL if a value passed to the function is NULL.

Example

The following expression returns the ASCII value for the first character of each value in the ITEMS column:

CHRCODE( ITEMS )

<table>
<thead>
<tr>
<th>ITEMS</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flashlight</td>
<td>70</td>
</tr>
<tr>
<td>Compass</td>
<td>67</td>
</tr>
<tr>
<td>Safety Knife</td>
<td>83</td>
</tr>
<tr>
<td>Depth/Pressure Gauge</td>
<td>68</td>
</tr>
<tr>
<td>Regulator System</td>
<td>82</td>
</tr>
</tbody>
</table>
COMPRESS

Compresses data using the zlib compression algorithm. The zlib compression algorithm is compatible with WinZip. Use the COMPRESS function before you send large amounts of data over a wide area network.

Syntax

COMPRESS( value )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>String datatype. Data that you want to compress.</td>
</tr>
</tbody>
</table>

Return Value

Compressed binary value of the input value.

NULL if the input is a null value.

Example

Your organization has an online order service. You want to send customer order data over a wide area network. The source contains a row that is 10 MB. You can compress the data in this row using COMPRESS. When you compress the data, you decrease the amount of data Informatica Cloud writes over the network. As a result, you may improve task performance.

CONCAT

Concatenates two strings. CONCAT converts all data to text before concatenating the strings. Alternatively, use the || string operator to concatenate strings. Using the || string operator instead of CONCAT improves performance when you run tasks.

Syntax

CONCAT( first_string, second_string )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>first_string</td>
<td>Required</td>
<td>Any datatype except Binary. The first part of the string you want to concatenate. You can enter any valid expression.</td>
</tr>
<tr>
<td>second_string</td>
<td>Required</td>
<td>Any datatype except Binary. The second part of the string you want to concatenate. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

String.

NULL if both string values are NULL.

Nulls

If one of the strings is NULL, CONCAT ignores it and returns the other string.
If both strings are NULL, CONCAT returns NULL.

Example

The following expression concatenates the names in the FIRST_NAME and LAST_NAME columns:

```
CONCAT( FIRST_NAME, LAST_NAME )
```

<table>
<thead>
<tr>
<th>FIRST_NAME</th>
<th>LAST_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Baer</td>
<td>JohnBaer</td>
</tr>
<tr>
<td>NULL</td>
<td>Campbell</td>
<td>Campbell</td>
</tr>
<tr>
<td>Bobbi</td>
<td>Apperley</td>
<td>BobbiApperley</td>
</tr>
<tr>
<td>Jason</td>
<td>Wood</td>
<td>JasonWood</td>
</tr>
<tr>
<td>Dan</td>
<td>Covington</td>
<td>DanCovington</td>
</tr>
<tr>
<td>Greg</td>
<td>NULL</td>
<td>Greg</td>
</tr>
<tr>
<td>100</td>
<td>200</td>
<td>100200</td>
</tr>
</tbody>
</table>

CONCAT does not add spaces to separate strings. If you want to add a space between two strings, you can write an expression with two nested CONCAT functions. For example, the following expression first concatenates a space on the end of the first name and then concatenates the last name:

```
CONCAT( CONCAT( FIRST_NAME, ' ' ), LAST_NAME )
```

<table>
<thead>
<tr>
<th>FIRST_NAME</th>
<th>LAST_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Baer</td>
<td>John Baer</td>
</tr>
<tr>
<td>NULL</td>
<td>Campbell</td>
<td>Campbell (includes leading space)</td>
</tr>
<tr>
<td>Bobbi</td>
<td>Apperley</td>
<td>Bobbi Apperley</td>
</tr>
<tr>
<td>Jason</td>
<td>Wood</td>
<td>Jason Wood</td>
</tr>
<tr>
<td>Dan</td>
<td>Covington</td>
<td>Dan Covington</td>
</tr>
<tr>
<td>Greg</td>
<td>NULL</td>
<td>Greg</td>
</tr>
<tr>
<td>100</td>
<td>200</td>
<td>100200</td>
</tr>
</tbody>
</table>

Use the CHR and CONCAT functions to concatenate a single quotation mark onto a string. The single quotation mark is the only character you cannot use inside a string literal. Consider the following example:

```
CONCAT( 'Joan', CONCAT( CHR(39), 's car' ))
```

The return value is:

```
Joan's car
```

**CONVERT_BASE**

Converts a number from one base value to another base value.

**Syntax**

```
CONVERT_BASE( value, source_base, dest_base )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>String datatype. Value you want to convert from one base to another base.</td>
</tr>
<tr>
<td>dest_base</td>
<td>Required</td>
<td>Numeric datatype. Base value you want to convert the data to. Minimum base is 2. Maximum base is 36.</td>
</tr>
</tbody>
</table>

**Return Value**

Numeric value.
Example

The following example converts 2222 from the decimal base value 10 to the binary base value 2:

\[
\text{CONVERT\_BASE( 2222, 10, 2 )}
\]

Informatica Cloud returns 100010101110.

\section*{COS}

Returns the cosine of a numeric value (expressed in radians).

\subsection*{Syntax}

\[
\text{COS( numeric\_value )}
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. Numeric data expressed in radians (degrees multiplied by pi divided by 180). Passes the values for which you want to calculate a cosine. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

\subsection*{Return Value}

Double value.

NULL if a value passed to the function is NULL.

Example

The following expression returns the cosine for all values in the DEGREES column:

\[
\text{COS( DEGREES * 3.14159265359 / 180 )}
\]

<table>
<thead>
<tr>
<th>DEGREES</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0</td>
</tr>
<tr>
<td>90</td>
<td>0.0</td>
</tr>
<tr>
<td>70</td>
<td>0.342020143325593</td>
</tr>
<tr>
<td>30</td>
<td>0.866025403784421</td>
</tr>
<tr>
<td>5</td>
<td>0.996194698091705</td>
</tr>
<tr>
<td>18</td>
<td>0.951056516295147</td>
</tr>
<tr>
<td>89</td>
<td>0.0174524064971813</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Tip

You can perform arithmetic on the values passed to COS before the function calculates the cosine. For example, you can convert the values in the column to radians before calculating the cosine, as follows:

\[
\text{COS( ARCS * 3.14159265359 / 180 )}
\]

\section*{COSH}

Returns the hyperbolic cosine of a numeric value (expressed in radians).
Syntax

\[ \text{COSH( numeric\_value )} \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. Numeric data expressed in radians (degrees multiplied by pi divided by 180). Passes the values for which you want to calculate the hyperbolic cosine. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

Double value.

NULL if a value passed to the function is NULL.

Example

The following expression returns the hyperbolic cosine for the values in the ANGLES column:

\[ \text{COSH( ANGLES )} \]

<table>
<thead>
<tr>
<th>ANGLES</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.54308063481524</td>
</tr>
<tr>
<td>2.897</td>
<td>9.0874465864177</td>
</tr>
<tr>
<td>3.66</td>
<td>19.435376920294</td>
</tr>
<tr>
<td>5.45</td>
<td>116.381231106176</td>
</tr>
<tr>
<td>0</td>
<td>1.0</td>
</tr>
<tr>
<td>0.345</td>
<td>1.06010513656773</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Tip

You can perform arithmetic on the values passed to \text{COSH} before the function calculates the hyperbolic cosine. For example:

\[ \text{COSH( MEASURES.ARCS / 360 )} \]

CRC32

Returns a 32-bit Cyclic Redundancy Check (CRC32) value. Use CRC32 to find data transmission errors. You can also use CRC32 if you want to verify that data stored in a file has not been modified.

Note: CRC32 can return the same output for different input strings. If you use CRC32 to generate keys, you may receive unexpected results.

Syntax

\[ \text{CRC32( value )} \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>String or Binary datatype. Passes the values you want to perform a redundancy check on. Input value is case sensitive. The case of the input value affects the return value. For example, CRC32(informatica) and CRC32 (Informatica) return different values.</td>
</tr>
</tbody>
</table>

Return Value

32-bit integer value.
Example

You want to read data from a source across a wide area network. You want to make sure the data has been modified during transmission. You can compute the checksum for the data in the file and store it along with the file. When Informatica Cloud reads the source data, Informatica Cloud can use CRC32 to compute the checksum and compare it to the stored value. If the two values are the same, the data has not been modified.

CUME

Returns a running total. A running total means CUME returns a total each time it adds a value. You can add a condition to filter rows out of the row set before calculating the running total.

Use CUME and similar functions, such as MOVINGAVG and MOVINGSUM, to simplify reporting by calculating running values.

Syntax

\[
\text{CUME( numeric\_value [, filter\_condition] )}
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. Passes the values for which you want to calculate a running total. You can enter any valid expression. You can create a nested expression to calculate a running total based on the results of the function as long as the result is a numeric value.</td>
</tr>
<tr>
<td>filter_condition</td>
<td>Optional</td>
<td>Limits the rows in the search. The filter condition must be a numeric value or evaluate to TRUE, FALSE, or NULL. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

Numeric value.

NULL if all values passed to the function are NULL, or if no rows are selected (for example, the filter condition evaluates to FALSE or NULL for all rows).

Nulls

If a value is NULL, CUME returns the running total for the previous row. However, if all values in the selected column are NULL, CUME returns NULL.

Examples

The following example returns the running total of the PERSONAL_SALES column:

\[
\text{CUME( PERSONAL\_SALES )}
\]

<table>
<thead>
<tr>
<th>PERSONAL_SALES</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>40000</td>
<td>40000</td>
</tr>
<tr>
<td>80000</td>
<td>120000</td>
</tr>
<tr>
<td>40000</td>
<td>160000</td>
</tr>
<tr>
<td>60000</td>
<td>220000</td>
</tr>
<tr>
<td>NULL</td>
<td>220000</td>
</tr>
<tr>
<td>50000</td>
<td>270000</td>
</tr>
</tbody>
</table>

Likewise, you can add values before calculating a running total:

\[
\text{CUME( CA\_SALES + OR\_SALES )}
\]

<table>
<thead>
<tr>
<th>CA_SALES</th>
<th>OR_SALES</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>40000</td>
<td>10000</td>
<td>50000</td>
</tr>
<tr>
<td>80000</td>
<td>50000</td>
<td>180000</td>
</tr>
</tbody>
</table>
DATE_COMPARE

Returns an integer indicating which of two dates is earlier. DATE_COMPARE returns an integer value rather than a date value.

Syntax

```
DATE_COMPARE( date1, date2 )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date1</td>
<td>Required</td>
<td>Date/Time datatype. The first date you want to compare. You can enter any valid expression as long as it evaluates to a date.</td>
</tr>
<tr>
<td>date2</td>
<td>Required</td>
<td>Date/Time datatype. The second date you want to compare. You can enter any valid expression as long as it evaluates to a date.</td>
</tr>
</tbody>
</table>

Return Value

-1 if the first date is earlier.
0 if the two dates are equal.
1 if the second date is earlier.
NULL if one of the date values is NULL.

Example

The following expression compares each date in the DATE_PROMISED and DATE_SHIPPED columns, and returns an integer indicating which date is earlier:

```
DATE_COMPARE( DATE_PROMISED, DATE_SHIPPED )
```

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 1 1997</td>
<td>Jan 13 1997</td>
<td>-1</td>
</tr>
<tr>
<td>Feb 1 1997</td>
<td>Feb 1 1997</td>
<td>0</td>
</tr>
<tr>
<td>Dec 22 1997</td>
<td>Dec 15 1997</td>
<td>1</td>
</tr>
<tr>
<td>Feb 29 1996</td>
<td>Apr 12 1996</td>
<td>-1 (Leap year)</td>
</tr>
<tr>
<td>NULL</td>
<td>Jan 6 1997</td>
<td>NULL</td>
</tr>
<tr>
<td>Jan 13 1997</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

DATE_DIFF

Returns the length of time between two dates. You can request the format to be years, months, days, hours, minutes, or seconds. Informatica Cloud subtracts the second date from the first date and returns the difference.
Syntax

\[
\text{DATE\_DIFF( date1, date2, format )}
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date1</td>
<td>Required</td>
<td>Date/Time datatype. Passes the values for the first date you want to compare. You can enter any valid expression.</td>
</tr>
<tr>
<td>date2</td>
<td>Required</td>
<td>Date/Time datatype. Passes the values for the second date you want to compare. You can enter any valid expression.</td>
</tr>
<tr>
<td>format</td>
<td>Required</td>
<td>Format string specifying the date or time measurement. You can specify years, months, days, hours, minutes, or seconds. You can specify only one part of the date, such as 'mm'. Enclose the format strings within single quotation marks. The format string is not case sensitive. For example, the format string 'mm' is the same as 'MM', 'Mm' or 'mM'.</td>
</tr>
</tbody>
</table>

Return Value

Double value. If \textit{date1} is later than \textit{date2}, the return value is a positive number. If \textit{date1} is earlier than \textit{date2}, the return value is a negative number.

0 if the dates are the same.

NULL if one (or both) of the date values is NULL.

Example

The following expressions return the number of hours between the DATE\_PROMISED and DATE\_SHIPPED columns:

\[
\begin{align*}
\text{DATE\_DIFF( DATE\_PROMISED, DATE\_SHIPPED, 'HH' )} \\
\text{DATE\_DIFF( DATE\_PROMISED, DATE\_SHIPPED, 'HH12' )} \\
\text{DATE\_DIFF( DATE\_PROMISED, DATE\_SHIPPED, 'HH24' )}
\end{align*}
\]

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 1 1997 12:00:00AM</td>
<td>Mar 29 1997 12:00:00PM</td>
<td>-2100</td>
</tr>
<tr>
<td>Mar 29 1997 12:00:00PM</td>
<td>Jan 1 1997 12:00:00AM</td>
<td>2100</td>
</tr>
<tr>
<td>NULL</td>
<td>Dec 10 1997 5:55:10PM</td>
<td>NULL</td>
</tr>
<tr>
<td>Dec 10 1997 5:55:10PM</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>Feb 19 2004 12:00:00PM</td>
<td>Feb 19 2005 12:00:00PM</td>
<td>-8784</td>
</tr>
</tbody>
</table>

The following expressions return the number of days between the DATE\_PROMISED and the DATE\_SHIPPED columns:

\[
\begin{align*}
\text{DATE\_DIFF( DATE\_PROMISED, DATE\_SHIPPED, 'D' )} \\
\text{DATE\_DIFF( DATE\_PROMISED, DATE\_SHIPPED, 'DD' )} \\
\text{DATE\_DIFF( DATE\_PROMISED, DATE\_SHIPPED, 'DDD' )} \\
\text{DATE\_DIFF( DATE\_PROMISED, DATE\_SHIPPED, 'DY' )} \\
\text{DATE\_DIFF( DATE\_PROMISED, DATE\_SHIPPED, 'DAY' )}
\end{align*}
\]

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 1 1997 12:00:00AM</td>
<td>Mar 29 1997 12:00:00PM</td>
<td>-87.5</td>
</tr>
<tr>
<td>Mar 29 1997 12:00:00PM</td>
<td>Jan 1 1997 12:00:00AM</td>
<td>87.5</td>
</tr>
<tr>
<td>NULL</td>
<td>Dec 10 1997 5:55:10PM</td>
<td>NULL</td>
</tr>
<tr>
<td>Dec 10 1997 5:55:10PM</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>Feb 19 2004 12:00:00PM</td>
<td>Feb 19 2005 12:00:00PM</td>
<td>-366</td>
</tr>
</tbody>
</table>
The following expressions return the number of months between the DATE_PROMISED and DATE_SHIPPED columns:

\[
\text{DATE_DIFF( DATE_PROMISED, DATE_SHIPPED, 'MM' )}
\]
\[
\text{DATE_DIFF( DATE_PROMISED, DATE_SHIPPED, 'MON' )}
\]
\[
\text{DATE_DIFF( DATE_PROMISED, DATE_SHIPPED, 'MONTH' )}
\]

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 1 1997 12:00:00AM</td>
<td>Mar 29 1997 12:00:00PM</td>
<td>-2.91935483870968</td>
</tr>
<tr>
<td>Mar 29 1997 12:00:00PM</td>
<td>Jan 1 1997 12:00:00AM</td>
<td>2.91935483870968</td>
</tr>
<tr>
<td>NULL</td>
<td>Dec 10 1997 5:55:10PM</td>
<td>NULL</td>
</tr>
<tr>
<td>Dec 10 1997 5:55:10PM</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>Feb 19 2004 12:00:00PM</td>
<td>Feb 19 2005 12:00:00PM</td>
<td>-12</td>
</tr>
</tbody>
</table>

The following expressions return the number of years between the DATE_PROMISED and DATE_SHIPPED columns:

\[
\text{DATE_DIFF( DATE_PROMISED, DATE_SHIPPED, 'Y' )}
\]
\[
\text{DATE_DIFF( DATE_PROMISED, DATE_SHIPPED, 'YY' )}
\]
\[
\text{DATE_DIFF( DATE_PROMISED, DATE_SHIPPED, 'YYY' )}
\]
\[
\text{DATE_DIFF( DATE_PROMISED, DATE_SHIPPED, 'YYYY' )}
\]

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 1 1997 12:00:00AM</td>
<td>Mar 29 1997 12:00:00PM</td>
<td>-0.24327956989247</td>
</tr>
<tr>
<td>Mar 29 1997 12:00:00PM</td>
<td>Jan 1 1997 12:00:00AM</td>
<td>0.24327956989247</td>
</tr>
<tr>
<td>NULL</td>
<td>Dec 10 1997 5:55:10PM</td>
<td>NULL</td>
</tr>
<tr>
<td>Dec 10 1997 5:55:10PM</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>Feb 19 2004 12:00:00PM</td>
<td>Feb 19 2005 12:00:00PM</td>
<td>-1</td>
</tr>
</tbody>
</table>

**DEC_BASE64**

Decodes a base 64 encoded value and returns a string with the binary data representation of the data.

**Syntax**

\[
\text{DEC_BASE64( value )}
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>String datatype. Data that you want to decode.</td>
</tr>
</tbody>
</table>

**Return Value**

Binary decoded value.

NULL if the input is a null value.

**Example**

You encoded MQSeries message IDs and wrote them to a flat file. You want to read data from the flat file source, including the MQSeries message IDs. You can use DEC_BASE64 to decode the IDs and convert them to their original binary value.
DECODE

Searches a column for a value that you specify. If the function finds the value, it returns a result value, which you define. You can build an unlimited number of searches within a DECODE function.

If you use DECODE to search for a value in a string column, you can either trim trailing blanks with the RTRIM function or include the blanks in the search string.

Syntax

```
DECODE( value, first_search, first_result [, second_search, second_result]...[,default] )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>Any datatype except Binary. Passes the values you want to search. You can enter any valid expression.</td>
</tr>
<tr>
<td>search</td>
<td>Required</td>
<td>Any value with the same datatype as the value argument. Passes the values for which you want to search. The search value must match the value argument. You cannot search for a portion of a value. Also, the search value is case sensitive. For example, if you want to search for the string 'Halogen Flashlight' in a particular column, you must enter 'Halogen Flashlight', not just 'Halogen'. If you enter 'Halogen', the search does not find a matching value. You can enter any valid expression.</td>
</tr>
<tr>
<td>result</td>
<td>Required</td>
<td>Any datatype except Binary. The value you want to return if the search finds a matching value. You can enter any valid expression.</td>
</tr>
<tr>
<td>default</td>
<td>Optional</td>
<td>Any datatype except Binary. The value you want to return if the search does not find a matching value. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

- **First_result** if the search finds a matching value.
- Default value if the search does not find a matching value.
- **NULL** if you omit the default argument and the search does not find a matching value.

Even if multiple conditions are met, Informatica Cloud returns the first matching result.

If the data contains multibyte characters and the DECODE expression compares string data, the return value depends on the code page of the Secure Agent that runs the task.

**DECODE and Datatypes**

When you use DECODE, the datatype of the return value is always the same as the datatype of the result with the greatest precision.

For example, you have the following expression:

```
DECODE ( CONST_NAME,
    'Five', 5,
    'Pythagoras', 1.414213562,
    'Archimedes', 3.141592654,
    'Pi', 3.141592654 )
```

The return values in this expression are 5, 1.414213562, and 3.141592654. The first result is an Integer, and the other results are Decimal. The Decimal datatype has greater precision than Integer. This expression always writes the result as a Decimal.

If at least one result is Double, the datatype of the return value is Double.
You cannot create a DECODE function with both string and numeric return values.

For example, the following expression is invalid because the return values include both string and numeric values:

```
DECODE ( CONST_NAME,
    'Five', 5,
    'Pythagoras', '1.414213562',
    'Archimedes', '3.141592654',
    'Pi', 3.141592654 )
```

**Example**

You might use DECODE in an expression that searches for a particular ITEM_ID and returns the ITEM_NAME:

```
DECODE( ITEM_ID, 10, 'Flashlight',
    14, 'Regulator',
    20, 'Knife',
    40, 'Tank',
    'NONE' )
```

<table>
<thead>
<tr>
<th>ITEM_ID</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Flashlight</td>
</tr>
<tr>
<td>14</td>
<td>Regulator</td>
</tr>
<tr>
<td>17</td>
<td>NONE</td>
</tr>
<tr>
<td>20</td>
<td>Knife</td>
</tr>
<tr>
<td>25</td>
<td>NONE</td>
</tr>
<tr>
<td>NULL</td>
<td>NONE</td>
</tr>
<tr>
<td>40</td>
<td>Tank</td>
</tr>
</tbody>
</table>

DECODE returns the default value of NONE for items 17 and 25 because the search values did not match the ITEM_ID. Also, DECODE returns NONE for the NULL ITEM_ID.

The following expression tests multiple columns and conditions, evaluated in a top to bottom order for TRUE or FALSE:

```
DECODE( TRUE,
    Var1 = 22, 'Variable 1 was 22!
    Var2 = 49, 'Variable 2 was 49!',
    Var1 < 23, 'Variable 1 was less than 23.',
    Var2 > 30, 'Variable 2 was more than 30.',
    'Variables were out of desired ranges.' )
```

<table>
<thead>
<tr>
<th>Var1</th>
<th>Var2</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>47</td>
<td>Variable 1 was less than 23.</td>
</tr>
<tr>
<td>22</td>
<td>49</td>
<td>Variable 1 was 22!</td>
</tr>
<tr>
<td>23</td>
<td>49</td>
<td>Variable 2 was 49!</td>
</tr>
<tr>
<td>24</td>
<td>27</td>
<td>Variables were out of desired ranges.</td>
</tr>
<tr>
<td>25</td>
<td>50</td>
<td>Variable 2 was more than 30.</td>
</tr>
</tbody>
</table>

**DECOMPRESS**

Decompresses data using the zlib compression algorithm. The zlib compression algorithm is compatible with WinZip. Use the DECOMPRESS function when you receive data over a wide area network.

**Syntax**

```
DECOMPRESS( value, precision)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>Binary datatype. Data that you want to decompress.</td>
</tr>
<tr>
<td>precision</td>
<td>Optional</td>
<td>Integer datatype.</td>
</tr>
</tbody>
</table>
Return Value

Decompressed binary value of the input value.

NULL if the input is a null value.

Example

Your organization has an online order service. You received compressed customer order data over a wide area network. You want to read the data using Informatica Cloud and load the data to a data warehouse. You can decompress each row of data using DECOMPRESS for the row. Informatica Cloud can then load the decompressed data to the target.

ENC_BASE64

Encodes data by converting binary data to string data using Multipurpose Internet Mail Extensions (MIME) encoding. Encode data when you want to store data in a database or file that does not allow binary data. You can also encode data to pass binary data in string format. The encoded data is approximately 33% longer than the original data. It displays as a set of random characters.

Syntax

```
ENC_BASE64( value )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>Binary or String datatype. Data that you want to encode.</td>
</tr>
</tbody>
</table>

Return Value

Encoded value.

NULL if the input is a null value.

Example

You want to read messages from an MQSeries source and write the data to a flat file target. You want to include the MQSeries message ID as part of the target data. However, the MsgID field is Binary, and the flat file target does not support binary data. Use ENC_BASE64 to encode the MsgID before Informatica Cloud writes the data to the target.

ERROR

Causes Informatica Cloud to skip a row and issue an error message, which you define. Informatica Cloud writes the skipped row and the error message into the error log file.

Use ERROR in expressions to validate data. Generally, you use ERROR within an IIF or DECODE function to set rules for skipping rows. You might use ERROR to keep null values from passing into a target.

You can also include ERROR in expression to handle transformation errors.
### Syntax

```plaintext
ERROR( string )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>string</code></td>
<td>Required</td>
<td>String value. The message you want to display when Informatica Cloud skips a row based on the expression containing the ERROR function. The string can be any length.</td>
</tr>
</tbody>
</table>

### Return Value

String.

### Example

The following example shows how to reference a mapping that calculates the average salary for employees in all departments of the organization, but skip negative values. The following expression nests the ERROR function in an IIF expression so that if Informatica Cloud finds a negative salary in the Salary column, it skips the row and displays an error:

```plaintext
IIF( SALARY < 0, ERROR ('Error. Negative salary found. Row skipped.', EMP_SALARY )
```

<table>
<thead>
<tr>
<th>SALARY</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>10000</td>
</tr>
<tr>
<td>-15000</td>
<td>'Error. Negative salary found. Row skipped.'</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>150000</td>
<td>150000</td>
</tr>
<tr>
<td>1005</td>
<td>1005</td>
</tr>
</tbody>
</table>

### EXP

Returns e raised to the specified power (exponent), where e=2.71828183. For example, EXP(2) returns 7.38905609893065. Use this function to analyze scientific and technical data. EXP is the reciprocal of the LN function, which returns the natural logarithm of a numeric value.

#### Syntax

```plaintext
EXP( exponent )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>exponent</code></td>
<td>Required</td>
<td>Numeric datatype. The value to which you want to raise e. The exponent in the equation e^value. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

### Return Value

Double value.

DOUBLE if a value passed as an argument to the function is NULL.

### Example

The following expression uses the values stored in the Numbers column as the exponent value:

```plaintext
EXP( NUMBERS )
```

<table>
<thead>
<tr>
<th>NUMBERS</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>22026.4657948067</td>
</tr>
<tr>
<td>-2</td>
<td>0.135335283236613</td>
</tr>
</tbody>
</table>
FLOOR

Returns the largest integer less than or equal to the numeric value you pass to this function. For example, if you pass 3.14 to FLOOR, the function returns 3. If you pass 3.98 to FLOOR, the function returns 3. Likewise, if you pass -3.17 to FLOOR, the function returns -4.

Syntax

```
FLOOR( numeric_value )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. You can enter any valid expression as long as it evaluates to numeric data.</td>
</tr>
</tbody>
</table>

Return Value

Integer if you pass a numeric value with declared precision between 0 and 28.

Double if you pass a numeric value with declared precision greater than 28.

NULL if a value passed to the function is NULL.

Example

The following expression returns the largest integer less than or equal to the values in the PRICE column:

```
FLOOR( PRICE )
```

<table>
<thead>
<tr>
<th>PRICE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>39.79</td>
<td>39</td>
</tr>
<tr>
<td>125.12</td>
<td>125</td>
</tr>
<tr>
<td>74.24</td>
<td>74</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>-100.99</td>
<td>-101</td>
</tr>
</tbody>
</table>

Tip

You can perform arithmetic on the values you pass to FLOOR. For example, to multiply a numeric value by 10 and then calculate the largest integer that is less than the product, you might write the function as follows:

```
FLOOR( UNIT_PRICE * 10 )
```

FV

Returns the future value of an investment, where you make periodic, constant payments and the investment earns a constant interest rate.
Syntax

\[ FV( \text{rate}, \text{terms}, \text{payment} [,\text{present value}, \text{type}] ) \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rate</td>
<td>Required</td>
<td>Numeric. Interest rate earned in each period. Expressed as a decimal number. Divide the percent rate by 100 to express it as a decimal number. Must be greater than or equal to 0.</td>
</tr>
<tr>
<td>terms</td>
<td>Required</td>
<td>Numeric. Number of periods or payments. Must be greater than 0.</td>
</tr>
<tr>
<td>payment</td>
<td>Required</td>
<td>Numeric. Payment amount due per period. Must be a negative number.</td>
</tr>
<tr>
<td>present value</td>
<td>Optional</td>
<td>Numeric. Current value of the investment. If you omit this argument, FV uses 0.</td>
</tr>
<tr>
<td>type</td>
<td>Optional</td>
<td>Integer. Timing of the payment. Enter 1 if payment is at the beginning of period. Enter 0 if payment is at the end of period. Default is 0. If you enter a value other than 0 or 1, Informatica Cloud treats the value as 1.</td>
</tr>
</tbody>
</table>

Return Value

Numeric.

Example

You deposit $2,000 into an account that earns 9% annual interest compounded monthly (monthly interest of 9%/12, or 0.75%). You plan to deposit $250 at the beginning of every month for the next 12 months. The following expression returns $5,337.96 as the account balance at the end of 12 months:

\[ FV(0.0075, 12, -250, -2000, \text{TRUE}) \]

Notes

To calculate interest rate earned in each period, divide the annual rate by the number of payments made in a year. The payment value and present value are negative because these are amounts that you pay.

---

**GET_DATE_PART**

Returns the specified part of a date as an integer value. For example, if you create an expression that returns the month portion of the date, and pass a date such as Apr 1 1997 00:00:00, GET_DATE_PART returns 4.

Syntax

\[ \text{GET_DATE_PART}( \text{date}, \text{format} ) \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>Required</td>
<td>Date/Time datatype. You can enter any valid expression.</td>
</tr>
<tr>
<td>format</td>
<td>Required</td>
<td>A format string specifying the portion of the date value you want to change. Enclose format strings within single quotation marks, for example, 'mm'. The format string is not case sensitive. Each format string returns the entire part of the date based on the default format of MM/DD/YYYY HH24:MI:SS.</td>
</tr>
<tr>
<td>Argument</td>
<td>Required/Optional</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example, if you pass the date Apr 1 1997 to GET_DATE_PART, the format strings 'Y', 'YY', 'YYY', or 'YYYY' all return 1997.</td>
</tr>
</tbody>
</table>

**Return Value**

Integer representing the specified part of the date.

NULL if a value passed to the function is NULL.

**Example**

The following expressions return the hour for each date in the DATE_SHIPPED column. 12:00:00AM returns 0 because the default date format is based on the 24 hour interval:

```sql
GET_DATE_PART( DATE_SHIPPED, 'HH' )
GET_DATE_PART( DATE_SHIPPED, 'HH12' )
GET_DATE_PART( DATE_SHIPPED, 'HH24' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar 13 1997 12:00:00AM</td>
<td>0</td>
</tr>
<tr>
<td>Sep 2 1997 2:00:01AM</td>
<td>2</td>
</tr>
<tr>
<td>Aug 22 1997 12:00:00PM</td>
<td>12</td>
</tr>
<tr>
<td>June 3 1997 11:30:44PM</td>
<td>23</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions return the day for each date in the DATE_SHIPPED column:

```sql
GET_DATE_PART( DATE_SHIPPED, 'D' )
GET_DATE_PART( DATE_SHIPPED, 'DD' )
GET_DATE_PART( DATE_SHIPPED, 'DDD' )
GET_DATE_PART( DATE_SHIPPED, 'DY' )
GET_DATE_PART( DATE_SHIPPED, 'DAY' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar 13 1997 12:00:00AM</td>
<td>13</td>
</tr>
<tr>
<td>June 3 1997 11:30:44PM</td>
<td>3</td>
</tr>
<tr>
<td>Aug 22 1997 12:00:00PM</td>
<td>22</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions return the month for each date in the DATE_SHIPPED column:

```sql
GET_DATE_PART( DATE_SHIPPED, 'MM' )
GET_DATE_PART( DATE_SHIPPED, 'MON' )
GET_DATE_PART( DATE_SHIPPED, 'MONTH' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar 13 1997 12:00:00AM</td>
<td>3</td>
</tr>
<tr>
<td>June 3 1997 11:30:44PM</td>
<td>6</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expression return the year for each date in the DATE_SHIPPED column:

```sql
GET_DATE_PART( DATE_SHIPPED, 'Y' )
GET_DATE_PART( DATE_SHIPPED, 'YY' )
GET_DATE_PART( DATE_SHIPPED, 'YYY' )
GET_DATE_PART( DATE_SHIPPED, 'YYYY' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar 13 1997 12:00:00AM</td>
<td>1997</td>
</tr>
<tr>
<td>June 3 1997 11:30:44PM</td>
<td>1997</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>
GREATEST

Returns the greatest value from a list of input values. Use this function to return the greatest string, date, or number. By default, the match is case sensitive.

Syntax

GREATEST( value1, [value2, ..., valueN], CaseFlag )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>Any datatype except Binary. Datatype must be compatible with other values. Value you want to compare against other values. You must enter at least one value argument. If the value is numeric, and other input values are numeric, all values use the highest precision possible. For example, if some values are Integer datatype and others are Double datatype, Informatica Cloud converts the values to Double.</td>
</tr>
<tr>
<td>CaseFlag</td>
<td>Optional</td>
<td>Must be an integer. Determines whether the arguments in this function are case sensitive. You can enter any valid expression. When CaseFlag is a number other than 0, the function is case sensitive. When CaseFlag is a null value or 0, the function is not case sensitive.</td>
</tr>
</tbody>
</table>

Return Value

value1 if it is the greatest of the input values, value2 if it is the greatest of the input values, and so on.

NULL if all the arguments are null.

Example

The following expression returns the greatest quantity of items ordered:

GREATEST( QUANTITY1, QUANTITY2, QUANTITY3 )

<table>
<thead>
<tr>
<th>QUANTITY1</th>
<th>QUANTITY2</th>
<th>QUANTITY3</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>756</td>
<td>27</td>
<td>756</td>
</tr>
<tr>
<td>5000</td>
<td>97</td>
<td>17</td>
<td>5000</td>
</tr>
<tr>
<td>120</td>
<td>1724</td>
<td>965</td>
<td>1724</td>
</tr>
</tbody>
</table>

IIF

Returns one of two values you specify, based on the results of a condition.
Syntax

\[
\text{IIF} ( \text{condition}, \text{value1} [,\text{value2}] )
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>condition</td>
<td>Required</td>
<td>The condition you want to evaluate. You can enter any valid expression that evaluates to TRUE or FALSE.</td>
</tr>
<tr>
<td>value1</td>
<td>Required</td>
<td>Any datatype except Binary. The value you want to return if the condition is TRUE. The return value is always the datatype specified by this argument. You can enter any valid expression, including another IIF expression.</td>
</tr>
<tr>
<td>value2</td>
<td>Optional</td>
<td>Any datatype except Binary. The value you want to return if the condition is FALSE. You can enter any valid expression, including another IIF expression.</td>
</tr>
</tbody>
</table>

Unlike conditional functions in some systems, the FALSE (\text{value2}) condition in the IIF function is not required. If you omit \text{value2}, the function returns the following when the condition is FALSE:

- 0 if \text{value1} is a Numeric datatype.
- Empty string if \text{value1} is a String datatype.
- NULL if \text{value1} is a Date/Time datatype.

For example, the following expression does not include a FALSE condition and \text{value1} is a string datatype so Informatica Cloud returns an empty string for each row that evaluates to FALSE:

\[
\text{IIF} ( \text{SALES} > 100, \text{EMP_NAME} )
\]

<table>
<thead>
<tr>
<th>SALES</th>
<th>EMP_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>John Smith</td>
<td>John Smith</td>
</tr>
<tr>
<td>50</td>
<td>Pierre Bleu</td>
<td>'' (empty string)</td>
</tr>
<tr>
<td>120</td>
<td>Sally Green</td>
<td>'' (empty string)</td>
</tr>
<tr>
<td>NULL</td>
<td>Greg Jones</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Return Value

\text{value1} if the condition is TRUE.

\text{value2} if the condition is FALSE.

For example, the following expression includes the FALSE condition NULL so Informatica Cloud returns NULL for each row that evaluates to FALSE:

\[
\text{IIF} ( \text{SALES} > 100, \text{EMP_NAME}, \text{NULL} )
\]

<table>
<thead>
<tr>
<th>SALES</th>
<th>EMP_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>John Smith</td>
<td>John Smith</td>
</tr>
<tr>
<td>50</td>
<td>Pierre Bleu</td>
<td>NULL</td>
</tr>
<tr>
<td>120</td>
<td>Sally Green</td>
<td>Sally Green</td>
</tr>
<tr>
<td>NULL</td>
<td>Greg Jones</td>
<td>NULL</td>
</tr>
</tbody>
</table>

If the data contains multibyte characters and the condition argument compares string data, the return value depends on the code page of the Secure Agent that runs the task.

IIF and Datatypes

When you use IIF, the datatype of the return value is the same as the datatype of the result with the greatest precision.

For example, you have the following expression:

\[
\text{IIF} ( \text{SALES} < 100, 1, .3333 )
\]

The TRUE result (1) is an integer and the FALSE result (.3333) is a decimal. The Decimal datatype has greater precision than Integer, so the datatype of the return value is always a Decimal.
When at least one result is Double, the datatype of the return value is Double.

**Special Uses of IIF**

Use nested IIF statements to test multiple conditions. The following example tests for various conditions and returns 0 if sales is 0 or negative:

```
IIF( SALES > 0, IIF( SALES < 50, SALARY1, IIF( SALES < 100, SALARY2, IIF( SALES < 200, SALARY3, BONUS))), 0 )
```

You can make this logic more readable by adding comments:

```
IIF( SALES > 0,
    --then test to see if sales is between 1 and 49:
    IIF( SALES < 50, 
        --then return SALARY1
        SALARY1,
        --else test to see if sales is between 50 and 99:
        IIF( SALES < 100, 
            --then return
            SALARY2,
            --else test to see if sales is between 100 and 199:
            IIF( SALES < 200, 
                --then return
                SALARY3,
                --else for sales over 199, return
                BONUS)
        ),
    --else for sales less than or equal to zero, return
    0)
```

**Use IIF in update strategies. For example:**

```
IIF( ISNULL( ITEM_NAME ), DD_REJECT, DD_INSERT)
```

**Alternative to IIF**

Use “DECODE” on page 224 instead of IIF in many cases. DECODE may improve readability. The following shows how you use DECODE instead of IIF using the first example from the previous section:

```
DECODE( TRUE,
    SALES > 0 and SALES < 50, SALARY1,
    SALES > 49 and SALES < 100, SALARY2,
    SALES > 99 and SALES < 200, SALARY3,
    SALES > 199, BONUS)
```

**IN**

Matches input data to a list of values. By default, the match is case sensitive.
Syntax

IN( valueToSearch, value1, [value2, ..., valueN,] CaseFlag )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>valueToSearch</td>
<td>Required</td>
<td>Can be a string, date, or numeric value. Input value you want to match against a comma-separated list of values.</td>
</tr>
<tr>
<td>value</td>
<td>Required</td>
<td>Can be a string, date, or numeric value. Comma-separated list of values you want to search for. Values can be columns. There is no maximum number of values you can list.</td>
</tr>
<tr>
<td>CaseFlag</td>
<td>Optional</td>
<td>Must be an integer. Determines whether the arguments in this function are case sensitive. You can enter any valid expression. When CaseFlag is a number other than 0, the function is case sensitive. When CaseFlag is a null value or 0, the function is not case sensitive.</td>
</tr>
</tbody>
</table>

Return Value

TRUE (1) if the input value matches the list of values.

FALSE (0) if the input value does not match the list of values.

NULL if the input is a null value.

Example

The following expression determines if the input value is a safety knife, chisel point knife, or medium titanium knife. The input values do not have to match the case of the values in the comma-separated list:

\[
\text{IN( ITEM\_NAME, 'Chisel Point Knife', 'Medium Titanium Knife', 'Safety Knife', 0 )}
\]

<table>
<thead>
<tr>
<th>ITEM_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stabilizing Vest</td>
<td>0 (FALSE)</td>
</tr>
<tr>
<td>Safety knife</td>
<td>1 (TRUE)</td>
</tr>
<tr>
<td>Medium Titanium knife</td>
<td>1 (TRUE)</td>
</tr>
<tr>
<td></td>
<td>NULL</td>
</tr>
</tbody>
</table>

INDEXOF

Finds the index of a value among a list of values. By default, the match is case sensitive.

Syntax

INDEXOF( valueToSearch, string1, [string2, ..., stringN,] CaseFlag )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>valueToSearch</td>
<td>Required</td>
<td>String datatype. Value you want to search for in the list of strings.</td>
</tr>
<tr>
<td>string</td>
<td>Required</td>
<td>String datatype. Comma-separated list of values you want to search against. Values can be in string format. There is no maximum number of values you can list. The value is case sensitive, unless you set MatchCase to 0.</td>
</tr>
<tr>
<td>CaseFlag</td>
<td>Required</td>
<td>Must be an integer. Determines whether the arguments in this function are case sensitive. You can enter any valid expression.</td>
</tr>
</tbody>
</table>
Argument | Required/Optional | Description
--- | --- | ---
 |  | When CaseFlag is a number other than 0, the function is case sensitive. When CaseFlag is a null value or 0, the function is not case sensitive.

**Return Value**

1 if the input value matches `string1`, 2 if the input value matches `string2`, and so on.

0 if the input value is not found.

NULL if the input is a null value.

**Example**

The following expression determines if values from the ITEM_NAME column match the first, second, or third string:

```
INDEXOF( ITEM_NAME, 'diving hood', 'flashlight', 'safety knife')
```

<table>
<thead>
<tr>
<th>ITEM_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety Knife</td>
<td>0</td>
</tr>
<tr>
<td>diving hood</td>
<td>1</td>
</tr>
<tr>
<td>Compass</td>
<td>0</td>
</tr>
<tr>
<td>safety knife</td>
<td>3</td>
</tr>
<tr>
<td>flashlight</td>
<td>2</td>
</tr>
</tbody>
</table>

Safety Knife returns a value of 0 because it does not match the case of the input value.

**INITCAP**

Capitalizes the first letter in each word of a string and converts all other letters to lowercase. Words are delimited by white space (a blank space, formfeed, newline, carriage return, tab, or vertical tab) and characters that are not alphanumeric. For example, if you pass the string ‘…THOMAS’, the function returns Thomas.

**Syntax**

```
INITCAP( string )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>Any datatype except Binary. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

**Return Value**

String. If the data contains multibyte characters, the return value depends on the code page of the Secure Agent that runs the task.

NULL if a value passed to the function is NULL.

**Example**

The following expression capitalizes all names in the FIRST_NAME column:

```
INITCAP( FIRST_NAME )
```

<table>
<thead>
<tr>
<th>FIRST_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ramona</td>
<td>Ramona</td>
</tr>
<tr>
<td>18-albert</td>
<td>18-Albert</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>
INSTR

Returns the position of a character set in a string, counting from left to right.

Syntax

```
INSTR( string, search_value [,start [,occurrence]] )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>string</code></td>
<td>Required</td>
<td>The string must be a character string. Passes the value you want to evaluate. You can enter any valid expression. The results of the expression must be a character string. If not, INSTR converts the value to a string before evaluating it.</td>
</tr>
<tr>
<td><code>search_value</code></td>
<td>Required</td>
<td>Any value. The search value is case sensitive. The set of characters you want to search for. The <code>search_value</code> must match a part of the string. For example, if you write <code>INSTR('Alfred Pope', 'Alfred Smith')</code> the function returns 0. You can enter any valid expression. If you want to search for a character string, enclose the characters you want to search for in single quotation marks, for example <code>'abc'</code>.</td>
</tr>
<tr>
<td><code>start</code></td>
<td>Optional</td>
<td>Must be an integer value. The position in the string where you want to start the search. You can enter any valid expression. The default is 1, meaning that INSTR starts the search at the first character in the string. If the start position is 0, INSTR searches from the first character in the string. If the start position is a positive number, INSTR locates the start position by counting from the beginning of the string. If the start position is a negative number, INSTR locates the start position by counting from the end of the string. If you omit this argument, the function uses the default value of 1.</td>
</tr>
<tr>
<td><code>occurrence</code></td>
<td>Optional</td>
<td>A positive integer greater than 0. You can enter any valid expression. If the search value appears more than once in the string, you can specify which occurrence you want to search for. For example, you would enter 2 to search for the second occurrence from the start position. If you omit this argument, the function uses the default value of 1, meaning that INSTR searches for the first occurrence of the search value. If you pass a decimal, Informatica Cloud rounds it to the nearest integer value. If you pass a negative integer or 0, the mapping fails when you run a workflow.</td>
</tr>
</tbody>
</table>

Return Value

Integer if the search is successful. Integer represents the position of the first character in the `search_value`, counting from left to right.

0 if the search is unsuccessful.

NULL if a value passed to the function is NULL.
Example

The following expression returns the position of the first occurrence of the letter ‘a’, starting at the beginning of each company name. Because the `search_value` argument is case sensitive, it skips the ‘A’ in ‘Blue Fin Aqua Center’, and returns the position for the ‘a’ in ‘Aqua’:

```
INSTR( COMPANY, 'a' )
```

<table>
<thead>
<tr>
<th>COMPANY</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue Fin Aqua Center</td>
<td>13</td>
</tr>
<tr>
<td>Maco Shark Shop</td>
<td>2</td>
</tr>
<tr>
<td>Scuba Gear</td>
<td>5</td>
</tr>
<tr>
<td>Frank's Dive Shop</td>
<td>3</td>
</tr>
<tr>
<td>VIP Diving Club</td>
<td>0</td>
</tr>
</tbody>
</table>

The following expression returns the position of the second occurrence of the letter ‘a’, starting at the beginning of each company name. Because the `search_value` argument is case sensitive, it skips the ‘A’ in ‘Blue Fin Aqua Center’, and returns 0:

```
INSTR( COMPANY, 'a', 1, 2 )
```

<table>
<thead>
<tr>
<th>COMPANY</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue Fin Aqua Center</td>
<td>0</td>
</tr>
<tr>
<td>Maco Shark Shop</td>
<td>8</td>
</tr>
<tr>
<td>Scuba Gear</td>
<td>9</td>
</tr>
<tr>
<td>Frank's Dive Shop</td>
<td>0</td>
</tr>
<tr>
<td>VIP Diving Club</td>
<td>0</td>
</tr>
</tbody>
</table>

The following expression returns the position of the second occurrence of the letter ‘a’ in each company name, starting from the last character in the company name. Because the `search_value` argument is case sensitive, it skips the ‘A’ in ‘Blue Fin Aqua Center’, and returns 0:

```
INSTR( COMPANY, 'a', -1, 2 )
```

<table>
<thead>
<tr>
<th>COMPANY</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue Fin Aqua Center</td>
<td>0</td>
</tr>
<tr>
<td>Maco Shark Shop</td>
<td>2</td>
</tr>
<tr>
<td>Scuba Gear</td>
<td>5</td>
</tr>
<tr>
<td>Frank's Dive Shop</td>
<td>0</td>
</tr>
<tr>
<td>VIP Diving Club</td>
<td>0</td>
</tr>
</tbody>
</table>

The following expression returns the position of the first character in the string ‘Blue Fin Aqua Center’ (starting from the last character in the company name):

```
INSTR( COMPANY, 'Blue Fin Aqua Center', -1, 1 )
```

<table>
<thead>
<tr>
<th>COMPANY</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue Fin Aqua Center</td>
<td>1</td>
</tr>
<tr>
<td>Maco Shark Shop</td>
<td>0</td>
</tr>
<tr>
<td>Scuba Gear</td>
<td>0</td>
</tr>
<tr>
<td>Frank's Dive Shop</td>
<td>0</td>
</tr>
<tr>
<td>VIP Diving Club</td>
<td>0</td>
</tr>
</tbody>
</table>

Using Nested INSTR

You can nest the INSTR function within other functions to accomplish more complex tasks.

The following expression evaluates a string, starting from the end of the string. The expression finds the last (rightmost) space in the string and then returns all characters to the left of it:

```
SUBSTR( CUST_NAME, 1, INSTR( CUST_NAME, ' ' ) )
```

<table>
<thead>
<tr>
<th>CUST_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATRICIA JONES</td>
<td>PATRICIA</td>
</tr>
<tr>
<td>MARY ELLEN SHAH</td>
<td>MARY ELLEN</td>
</tr>
</tbody>
</table>

The following expression removes the character '#' from a string:

```
SUBSTR( CUST_ID, INSTR(CUST_ID, '#')+1 )
```

<table>
<thead>
<tr>
<th>CUST_ID</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID#33</td>
<td>ID33</td>
</tr>
<tr>
<td>#A3577</td>
<td>A3577</td>
</tr>
</tbody>
</table>
IS_DATE

Returns whether a string value is a valid date. A valid date is any string in the default date format of MM/DD/YYYY HH24:MI:SS. If the strings you want to test are not in the default date format, use the TO_DATE format strings to specify the date format. If the strings passed to IS_DATE do not match the format string specified, the function returns FALSE (0). If the strings match the format string, the function returns TRUE (1).

IS_DATE evaluates strings and returns an integer value.

The target column for an IS_DATE expression must be String or Numeric datatype.

You might use IS_DATE to test or filter data in a flat file before writing it to a target.

Use the RR format string with IS_DATE instead of the YY format string. In most cases, the two format strings return the same values, but there are some unique cases where YY returns incorrect results. For example, the expression IS_DATE('02/29/00', 'YY') is internally computed as IS_DATE(02/29/1900 00:00:00), which returns false. However, Informatica Cloud computes the expression IS_DATE('02/29/00', 'RR') as IS_DATE(02/29/2000 00:00:00), which returns TRUE. In the first case, year 1900 is not a leap year, so there is no February 29th.

Note: IS_DATE uses the same format strings as TO_DATE.

Syntax

IS_DATE( value [, format] )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>Must be a string datatype. Passes the rows you want to evaluate. You can enter any valid expression.</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>Enter a valid TO_DATE format string. The format string must match the parts of the string argument. For example, if you pass the string 'Mar 15 1997 12:43:10AM', you must use the format string 'MON DD YYYY HH12:MI:SSAM'. If you omit the format string, the string value must be in the default date of MM/DD/YYYY HH24:MI:SS.</td>
</tr>
</tbody>
</table>

Return Value

TRUE (1) if the row is a valid date.
FALSE (0) if the row is not a valid date.
NULL if a value in the expression is NULL or if the format string is NULL.

Warning: The format of the IS_DATE string must match the format string, including any date separators. If it does not, Informatica Cloud might return inaccurate values or skip the row.

Example

The following expression checks the INVOICE_DATE column for valid dates:

```
IS_DATE( INVOICE_DATE )
```

This expression returns data similar to the following:

<table>
<thead>
<tr>
<th>INVOICE_DATE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>'180'</td>
<td>0 (FALSE)</td>
</tr>
<tr>
<td>'04/01/98'</td>
<td>0 (FALSE)</td>
</tr>
</tbody>
</table>
The following IS_DATE expression specifies a format string of 'YYYY/MM/DD':

\[ \text{IS_DATE( INVOICE\_DATE, 'YYYY/MM/DD' )} \]

If the string value does not match this format, IS_DATE returns FALSE:

<table>
<thead>
<tr>
<th>INVOICE_DATE</th>
<th>RETURN_VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>'180'</td>
<td>0 (FALSE)</td>
</tr>
<tr>
<td>'04/01/98'</td>
<td>0 (FALSE)</td>
</tr>
<tr>
<td>'1998/01/12'</td>
<td>1 (TRUE)</td>
</tr>
<tr>
<td>'1998/11/21 00:00:13'</td>
<td>0 (FALSE)   (February does not have 31 days)</td>
</tr>
<tr>
<td>'1998/02/31'</td>
<td>0 (FALSE)</td>
</tr>
<tr>
<td>'John Smith'</td>
<td>0 (FALSE)</td>
</tr>
</tbody>
</table>

The following example shows how you use IS_DATE to test data before using TO_DATE to convert the strings to dates. This expression checks the values in the INVOICE_DATE column and converts each valid date to a date value. If the value is not a valid date, Informatica Cloud returns ERROR and skips the row.

This example returns a Date/Time value. Therefore, the target column for the expression needs to be Date/Time:

\[ \text{IIF( IS\_DATE ( INVOICE\_DATE, 'YYYY/MM/DD' ), \text{TO\_DATE( INVOICE\_DATE )}, \text{ERROR('Not a valid date')})} \]

### IS\_NUMBER

Returns whether a string is a valid number.

A valid number consists of the following parts:

- Optional space before the number
- Optional sign (+/-)
- One or more digits with an optional decimal point
- Optional scientific notation, such as the letter ‘e’ or ‘E’ (and the letter ‘d’ or ‘D’ on Windows) followed by an optional sign (+/-), followed by one or more digits
- Optional white space following the number

The following numbers are all valid:

- `100`
- `+100`
- `-100`
- `'3.45e+32'`
- `'+3.45e+32'`
- `'3.45d+32'` (Windows only)
- `'+3.45D+32'` (Windows only)
- `.6804`

The target column for an IS\_NUMBER expression must be a String or Numeric datatype.

You might use IS\_NUMBER to test or filter data in a flat file before writing it to a target.
**Syntax**

```sql
IS_NUMBER( value )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>Must be a String datatype. Passes the rows you want to evaluate. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

**Return Value**

- TRUE (1) if the row is a valid number.
- FALSE (0) if the row is not a valid number.
- NULL if a value in the expression is NULL.

**Example**

The following expression checks the ITEM_PRICE column for valid numbers:

```sql
IS_NUMBER( ITEM_PRICE )
```

<table>
<thead>
<tr>
<th>ITEM_PRICE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'123.00'</td>
<td>1 (True)</td>
</tr>
<tr>
<td>'-3.45e+3'</td>
<td>1 (True)</td>
</tr>
<tr>
<td>'-3.45d-3'</td>
<td>1 (True - Windows only)</td>
</tr>
<tr>
<td>'-3.45D-3'</td>
<td>0 (False - UNIX only)</td>
</tr>
<tr>
<td>'3.45E-'</td>
<td>0 (False) Incomplete number</td>
</tr>
<tr>
<td>''</td>
<td>0 (False) Consists entirely of whitespace</td>
</tr>
<tr>
<td>'+123abc'</td>
<td>0 (False)</td>
</tr>
<tr>
<td>' 123'</td>
<td>1 (True) Leading whitespace</td>
</tr>
<tr>
<td>'123 '</td>
<td>1 (True) Trailing whitespace</td>
</tr>
<tr>
<td>'ABC'</td>
<td>0 (False)</td>
</tr>
<tr>
<td>'-ABC'</td>
<td>0 (False)</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Use IS_NUMBER to test data before using one of the numeric conversion functions, such as TO_FLOAT. For example, the following expression checks the values in the ITEM_PRICE column and converts each valid number to a double-precision floating point value. If the value is not a valid number, Informatica Cloud returns 0.00:

```sql
IIF( IS_NUMBER( ITEM_PRICE ), TO_FLOAT( ITEM_PRICE ), 0.00 )
```

<table>
<thead>
<tr>
<th>ITEM_PRICE</th>
<th>RETURN VALUE</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>'123.00'</td>
<td>123</td>
<td></td>
</tr>
<tr>
<td>'-3.45e+3'</td>
<td>-3450</td>
<td></td>
</tr>
<tr>
<td>'3.45E-3'</td>
<td>0.00345</td>
<td></td>
</tr>
<tr>
<td>''</td>
<td>0.00</td>
<td>Consists entirely of whitespace</td>
</tr>
<tr>
<td>'+123abc'</td>
<td>0.00</td>
<td>Empty string</td>
</tr>
<tr>
<td>' 123ABC'</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>'ABC'</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>'-ABC'</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td></td>
</tr>
</tbody>
</table>

**IS_SPACES**

Returns whether a string value consists entirely of spaces. A space is a blank space, a formfeed, a newline, a carriage return, a tab, or a vertical tab.

IS_SPACES evaluates an empty string as FALSE because there are no spaces. To test for an empty string, use LENGTH.
Syntax

```
IS_SPACES( value )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>Must be a string datatype. Passes the rows you want to evaluate. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

TRUE (1) if the row consists entirely of spaces.

FALSE (0) if the row contains data.

NULL if a value in the expression is NULL.

Example

The following expression checks the ITEM_NAME column for rows that consist entirely of spaces:

```
IS_SPACES( ITEM_NAME )
```

<table>
<thead>
<tr>
<th>ITEM_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flashlight</td>
<td>0 (False)</td>
</tr>
<tr>
<td>Regulator system</td>
<td>1 (True)</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>''</td>
<td>0 (FALSE) (Empty string does not contain spaces.)</td>
</tr>
</tbody>
</table>

Tip

Use IS_SPACES to avoid writing spaces to a character column in a target table. For example, if you want to write customer names to a fixed length CHAR(5) column in a target table, you might want to write '00000' instead of spaces. You would create an expression similar to the following:

```
IIF( IS_SPACES( CUST_NAMES ), '00000', CUST_NAMES )
```

**ISNULL**

Returns whether a value is NULL. ISNULL evaluates an empty string as FALSE.

**Note:** To test for empty strings, use LENGTH.

Syntax

```
ISNULL( value )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>Any datatype except Binary. Passes the rows you want to evaluate. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

TRUE (1) if the value is NULL.

FALSE (0) if the value is not NULL.
Example

The following example checks for null values in the items table:

```sql
ISNULL( ITEM_NAME )
```

<table>
<thead>
<tr>
<th>ITEM_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flashlight</td>
<td>0 (FALSE)</td>
</tr>
<tr>
<td>NULL</td>
<td>1 (TRUE)</td>
</tr>
<tr>
<td>Regulator system</td>
<td>0 (FALSE)</td>
</tr>
</tbody>
</table>
| ''                 | 0 (FALSE)    | Empty string is not NULL

LAST_DAY

Returns the date of the last day of the month for each date in a column.

Syntax

```sql
LAST_DAY( date )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>Required</td>
<td>Date/Time datatype. Passes the dates for which you want to return the last day of the month. You can enter any valid expression that evaluates to a date.</td>
</tr>
</tbody>
</table>

Return Value

Date. The last day of the month for that date value you pass to this function.

NULL if a value in the selected column is NULL.

Null

If a value is NULL, LAST_DAY ignores the row. However, if all values passed from the column are NULL, LAST_DAY returns NULL.

Example

The following expression returns the last day of the month for each date in the ORDER_DATE column:

```sql
LAST_DAY( ORDER_DATE )
```

<table>
<thead>
<tr>
<th>ORDER_DATE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apr 1 1998 12:00:00AM</td>
<td>Apr 30 1998 12:00:00AM</td>
</tr>
<tr>
<td>Jan 6 1998 12:00:00AM</td>
<td>Jan 31 1998 12:00:00AM</td>
</tr>
<tr>
<td>Feb 2 1996 12:00:00AM</td>
<td>Feb 29 1996 12:00:00AM (Leap year)</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>Jul 31 1998 12:00:00AM</td>
<td>Jul 31 1998 12:00:00AM</td>
</tr>
</tbody>
</table>

You can nest TO_DATE to convert string values to a date. TO_DATE always includes time information. If you pass a string that does not have a time value, the date returned will include the time 00:00:00.

The following example returns the last day of the month for each order date in the same format as the string:

```sql
LAST_DAY( TO_DATE( ORDER_DATE, 'DD-MON-YY' ) )
```

<table>
<thead>
<tr>
<th>ORDER_DATE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'18-NOV-98'</td>
<td>Nov 30 1998 00:00:00</td>
</tr>
<tr>
<td>'29-APR-98'</td>
<td>Apr 30 1998 00:00:00</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>'18-FEB-96'</td>
<td>Feb 29 1996 00:00:00 (Leap year)</td>
</tr>
</tbody>
</table>

242    Appendix B: Functions
**LEAST**

Returns the smallest value from a list of input values. By default, the match is case sensitive.

**Syntax**

\[
\text{LEAST}( \text{value}_1, \ldots, \text{value}_N, \text{CaseFlag} )
\]

**Argument**

<table>
<thead>
<tr>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Any datatype except Binary. Datatype must be compatible with other values. Value you want to compare against other values. You must enter at least one value argument. If the value is Numeric, and other input values are of other numeric datatypes, all values use the highest precision possible. For example, if some values are of the Integer datatype and others are of the Double datatype, Informatica Cloud converts the values to Double.</td>
</tr>
<tr>
<td>CaseFlag</td>
<td>Must be an integer. Determines whether the arguments in this function are case sensitive. You can enter any valid expression. When CaseFlag is a number other than 0, the function is case sensitive. When CaseFlag is a null value or 0, the function is not case sensitive.</td>
</tr>
</tbody>
</table>

**Return Value**

\[
\text{value}_1 \text{ if it is the smallest of the input values, } \text{value}_2 \text{ if it is the smallest of the input values, and so on.}
\]

NULL if all the arguments are null.

**Example**

The following expression returns the smallest quantity of items ordered:

\[
\text{LEAST}( \text{QUANTITY}_1, \text{QUANTITY}_2, \text{QUANTITY}_3 )
\]

<table>
<thead>
<tr>
<th>QUANTITY1</th>
<th>QUANTITY2</th>
<th>QUANTITY3</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>756</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>5000</td>
<td>97</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>120</td>
<td>1724</td>
<td>965</td>
<td>120</td>
</tr>
</tbody>
</table>

**LENGTH**

Returns the number of characters in a string, including trailing blanks.

**Syntax**

\[
\text{LENGTH}( \text{string} )
\]

**Argument**

<table>
<thead>
<tr>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>String datatype. The strings you want to evaluate. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

**Return Value**

Integer representing the length of the string.

NULL if a value passed to the function is NULL.
Example

The following expression returns the length of each customer name:

\[
\text{LENGTH( CUSTOMER\_NAME )}
\]

<table>
<thead>
<tr>
<th>CUSTOMER_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bernice Davis</td>
<td>13</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>John Baer</td>
<td>9</td>
</tr>
<tr>
<td>Greg Brown</td>
<td>10</td>
</tr>
</tbody>
</table>

Tips

Use \text{LENGTH} to test for empty string conditions. If you want to find fields in which customer name is empty, use an expression such as:

\[
\text{IIF( LENGTH( CUSTOMER\_NAME ) = 0, 'EMPTY STRING' )}
\]

To test for a null field, use \text{ISNULL}. To test for spaces, use \text{IS\_SPACES}.

\textbf{LN}

Returns the natural logarithm of a numeric value. For example, \text{LN(3)} returns 1.098612. You usually use this function to analyze scientific data rather than business data.

This function is the reciprocal of the function \text{EXP}.

Syntax

\[
\text{LN( numeric\_value )}
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. It must be a positive number, greater than 0. Passes the values for which you want to calculate the natural logarithm. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

Double value.

\text{NULL} if a value passed to the function is \text{NULL}.

Example

The following expression returns the natural logarithm for all values in the NUMBERS column:

\[
\text{LN( NUMBERS )}
\]

<table>
<thead>
<tr>
<th>NUMBERS</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2.302585092994</td>
</tr>
<tr>
<td>125</td>
<td>4.8283137637302</td>
</tr>
<tr>
<td>0.96</td>
<td>-0.04082199452026</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>-90</td>
<td>None. (Informatica Cloud writes the row to the error log file.)</td>
</tr>
<tr>
<td>0</td>
<td>None. (Informatica Cloud writes the row to the error log file.)</td>
</tr>
</tbody>
</table>

Note: When you pass a negative number or 0, Informatica Cloud writes the row into the error log file. The \text{numeric\_value} must be a positive number greater than 0.
LOG

Returns the logarithm of a numeric value. Most often, you use this function to analyze scientific data rather than business data.

Syntax

\[ \text{LOG}( \text{base}, \text{exponent} ) \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>Required</td>
<td>The base of the logarithm. Must be a positive numeric value other than 0 or 1. Any valid expression that evaluates to a positive number other than 0 or 1.</td>
</tr>
<tr>
<td>exponent</td>
<td>Required</td>
<td>The exponent of the logarithm. Must be a positive numeric value greater than 0. Any valid expression that evaluates to a positive number greater than 0.</td>
</tr>
</tbody>
</table>

Return Value

Double value.

NULL if a value passed to the function is NULL.

Example

The following expression returns the logarithm for all values in the NUMBERS column:

\[ \text{LOG}( \text{BASE}, \text{EXponent} ) \]

<table>
<thead>
<tr>
<th>BASE</th>
<th>EXponent</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>.09</td>
<td>10</td>
<td>-0.956244644696599</td>
</tr>
<tr>
<td>NULL</td>
<td>18</td>
<td>NULL</td>
</tr>
<tr>
<td>35.78</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>-9</td>
<td>18</td>
<td>NULL</td>
</tr>
</tbody>
</table>

None. (Informatica Cloud writes the row to the error log file.)

<table>
<thead>
<tr>
<th>BASE</th>
<th>EXponent</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>NULL</td>
</tr>
<tr>
<td>10</td>
<td>-2</td>
<td>NULL</td>
</tr>
</tbody>
</table>

None. (Informatica Cloud writes the row to the error log file.)

If you pass a negative number, 0, or 1 as a base value, or if you pass a negative value for the exponent, Informatica Cloud writes the row to the error log file.

LOWER

Converts uppercase string characters to lowercase.
**Syntax**

```
LOWER( string )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>Any string value. The argument passes the string values that you want to return as lowercase. You can enter any valid expression that evaluates to a string.</td>
</tr>
</tbody>
</table>

**Return Value**

Lowercase character string. If the data contains multibyte characters, the return value depends on the code page of the Secure Agent that runs the task.

NULL if a value in the selected column is NULL.

**Example**

The following expression returns all first names to lowercase:

```
LOWER( FIRST_NAME )
```

<table>
<thead>
<tr>
<th>FIRST_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>antonia</td>
<td>antonia</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>THOMAS</td>
<td>thomas</td>
</tr>
<tr>
<td>Pierre</td>
<td>pierre</td>
</tr>
<tr>
<td>BERNICE</td>
<td>bernice</td>
</tr>
</tbody>
</table>

**LPAD**

Adds a set of blanks or characters to the beginning of a string to set the string to a specified length.

**Syntax**

```
LPAD( first_string, length [,second_string] )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>first_string</td>
<td>Required</td>
<td>Can be a character string. The strings you want to change. You can enter any valid expression.</td>
</tr>
<tr>
<td>length</td>
<td>Required</td>
<td>Must be a positive integer literal. This argument specifies the length you want each string to be. When <code>length</code> is a negative number, RPAD returns NULL.</td>
</tr>
<tr>
<td>second_string</td>
<td>Optional</td>
<td>Can be any string value. The characters you want to append to the left-side of the <code>first_string</code> values. You can enter any valid expression. You can enter a specific string literal. However, enclose the characters you want to add to the beginning of the string within single quotation marks, as in 'abc'. This argument is case sensitive. If you omit the <code>second_string</code>, the function pads the beginning of the first string with blanks.</td>
</tr>
</tbody>
</table>

**Return Value**

String of the specified length.

NULL if a value passed to the function is NULL or if `length` is a negative number.
Example

The following expression standardizes numbers to six digits by padding them with leading zeros.

```sql
LPAD( PART_NUM, 6, '0')
```

<table>
<thead>
<tr>
<th>PART_NUM</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>702</td>
<td>000702</td>
</tr>
<tr>
<td>1</td>
<td>000001</td>
</tr>
<tr>
<td>0553</td>
<td>000553</td>
</tr>
<tr>
<td>484834</td>
<td>484834</td>
</tr>
</tbody>
</table>

LPAD counts the length from left to right. If the first string is longer than the length, LPAD truncates the string from right to left. For example, LPAD('alphabetical', 5, 'x') returns the string 'alpha'.

If the second string is longer than the total characters needed to return the specified length, LPAD uses a portion of the second string:

```sql
LPAD( ITEM_NAME, 16, '*..*')
```

<table>
<thead>
<tr>
<th>ITEM_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flashlight</td>
<td>*..*Flashlight</td>
</tr>
<tr>
<td>Compass</td>
<td>*..*Compass</td>
</tr>
<tr>
<td>Regulator System</td>
<td>Regulator System</td>
</tr>
<tr>
<td>Safety Knife</td>
<td>*..*Safety Knife</td>
</tr>
</tbody>
</table>

The following expression shows how LPAD handles negative values for the length argument for each row in the ITEM_NAME column:

```sql
LPAD( ITEM_NAME, -5, '.')
```

<table>
<thead>
<tr>
<th>ITEM_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flashlight</td>
<td>NULL</td>
</tr>
<tr>
<td>Compass</td>
<td>NULL</td>
</tr>
<tr>
<td>Regulator System</td>
<td>NULL</td>
</tr>
</tbody>
</table>

LTRIM

Removes blanks or characters from the beginning of a string. You can use LTRIM with IIF or DECODE in an expression to avoid spaces in a target table.

If you do not specify a trim_set parameter in the expression, LTRIM removes only single-byte spaces.

If you use LTRIM to remove characters from a string, LTRIM compares the trim_set to each character in the string argument, character-by-character, starting with the left side of the string. If the character in the string matches any character in the trim_set, LTRIM removes it. LTRIM continues comparing and removing characters until it fails to find a matching character in the trim_set. Then it returns the string, which does not include matching characters.

Syntax

```sql
LTRIM( string [, trim_set] )
```

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>Any string value. Passes the strings you want to modify. You can enter any valid expression. Use operators to perform comparisons or concatenate strings before removing characters from the beginning of a string.</td>
</tr>
<tr>
<td>trim_set</td>
<td>Optional</td>
<td>Any string value. Passes the characters you want to remove from the beginning of the first string. You can enter any valid expression. You can also enter a character string.</td>
</tr>
</tbody>
</table>
However, you must enclose the characters you want to remove from the beginning of the string within single quotation marks, for example, 'abc'. If you omit the second string, the function removes any blanks from the beginning of the string. LTRIM is case sensitive. For example, if you want to remove the 'A' character from the string 'Alfredo', you would enter 'A', not 'a'.

Return Value

String. The string values with the specified characters in the trim_set argument removed.

NULL if a value passed to the function is NULL. If the trim_set is NULL, the function returns NULL.

Example

The following expression removes the characters 'S' and '.' from the strings in the LAST_NAME column:

\[
\text{LTRIM( LAST_NAME, 'S.' )}
\]

<table>
<thead>
<tr>
<th>LAST_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nelson</td>
<td>Nelson</td>
</tr>
<tr>
<td>Osborne</td>
<td>Osborne</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>S. MacDonald</td>
<td>MacDonald</td>
</tr>
<tr>
<td>Sawyer</td>
<td>sawyer</td>
</tr>
<tr>
<td>H. Bender</td>
<td>H. Bender</td>
</tr>
<tr>
<td>Steadman</td>
<td>teadman</td>
</tr>
</tbody>
</table>

LTRIM removes 'S.' from S. MacDonald and the 'S' from both Sawyer and Steadman, but not the period from H. Bender. This is because LTRIM searches, character-by-character, for the set of characters you specify in the trim_set argument. If the first character in the string matches the first character in the trim_set, LTRIM removes it. Then LTRIM looks at the second character in the string. If it matches the second character in the trim_set, LTRIM removes it, and so on. When the first character in the string does not match the corresponding character in the trim_set, LTRIM returns the string and evaluates the next row.

In the example of H. Bender, H does not match either character in the trim_set argument, so LTRIM returns the string in the LAST_NAME column and moves to the next row.

Tips

Use RTRIM and LTRIM with || or CONCAT to remove leading and trailing blanks after you concatenate two strings. You can also remove multiple sets of characters by nesting LTRIM. For example, if you want to remove leading blanks and the character 'T' from a column of names, you might create an expression similar to the following:

\[
\text{LTRIM( LTRIM( NAMES ), 'T' )}
\]

**MAKE_DATE_TIME**

Returns the date and time based on the input values.
Syntax

MAKE_DATE_TIME( year, month, day, hour, minute, second )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>Required</td>
<td>Numeric datatype. Positive integer.</td>
</tr>
<tr>
<td>month</td>
<td>Required</td>
<td>Numeric datatype. Positive integer between 1 and 12 (January=1 and December=12).</td>
</tr>
<tr>
<td>day</td>
<td>Required</td>
<td>Numeric datatype. Positive integer between 1 and 31 (except for the months that have less than 31 days: February, April, June, September, and November).</td>
</tr>
<tr>
<td>hour</td>
<td>Optional</td>
<td>Numeric datatype. Positive integer between 0 and 24 (where 0=12AM, 12=12PM, and 24 =12AM).</td>
</tr>
<tr>
<td>minute</td>
<td>Optional</td>
<td>Numeric datatype. Positive integer between 0 and 59.</td>
</tr>
<tr>
<td>second</td>
<td>Optional</td>
<td>Numeric datatype. Positive integer between 0 and 59.</td>
</tr>
</tbody>
</table>

Return Value

Date as MM/DD/YYYY HH24:MI:SS.

Example

The following expression creates a date and time from the source columns:

MAKE_DATE_TIME( SALE_YEAR, SALE_MONTH, SALE_DAY, SALE_HOUR, SALE_MIN, SALE_SEC )

<table>
<thead>
<tr>
<th>SALE_YR</th>
<th>SALE_MTH</th>
<th>SALE_DAY</th>
<th>SALE_HR</th>
<th>SALE_MIN</th>
<th>SALE_SEC</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>10</td>
<td>27</td>
<td>8</td>
<td>36</td>
<td>22</td>
<td>10/27/2002 08:36:22</td>
</tr>
<tr>
<td>2000</td>
<td>6</td>
<td>15</td>
<td>15</td>
<td>17</td>
<td></td>
<td>06/15/2000 15:17:00</td>
</tr>
<tr>
<td>2003</td>
<td>1</td>
<td>3</td>
<td>12</td>
<td>22</td>
<td>45</td>
<td>01/03/2003 01:22:45</td>
</tr>
<tr>
<td>04</td>
<td>3</td>
<td>30</td>
<td>12</td>
<td>5</td>
<td>10</td>
<td>03/30/2004 12:05:10</td>
</tr>
<tr>
<td>99</td>
<td>12</td>
<td>12</td>
<td>5</td>
<td>16</td>
<td></td>
<td>12/12/1999?? 05:??:16</td>
</tr>
</tbody>
</table>

MD5

Calculates the checksum of the input value. The function uses Message-Digest algorithm 5 (MD5). MD5 is a one-way cryptographic hash function with a 128-bit hash value. It calculates a unique value for each input. Use MD5 to verify data integrity.

Syntax

MD5( value )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>String or Binary datatype. Value for which you want to calculate checksum. The case of the input value affects the return value. For example, MD5(informatica) and MD5(informatica) return different values.</td>
</tr>
</tbody>
</table>
Return Value

Unique 32-character string of hexadecimal digits 0-9 and a-f.
NULL if the input is a null value.

Example

You want to write changed data to a database. You can use the MD5 function to generate a unique checksum value for a row of data each time you read data from a source. When you run new sessions to read data from the same source, you can compare the previously generated checksum value against new checksum values. You can then write rows with new checksum values to the target. Those rows represent data that is changed in the source.

Tip

You can use the return value as a hash key.

METAPHONE

Encodes string values. You can specify the length of the string that you want to encode.

METAPHONE encodes characters of the English language alphabet (A-Z). It encodes both uppercase and lowercase letters in uppercase.

METAPHONE encodes characters according to the following list of rules:

- Skips vowels (A, E, I, O, and U) unless one of them is the first character of the input string. METAPHONE('CAR') returns 'KR' and METAPHONE('AAR') returns 'AR'.
- Uses special encoding guidelines.

The following table lists the METAPHONE encoding guidelines:

**Table 13. METAPHONE Encoding Guidelines**

<table>
<thead>
<tr>
<th>Input</th>
<th>Returns</th>
<th>Condition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>n/a</td>
<td>When it follows M.</td>
<td>METAPHONE ('Lamb') returns LM.</td>
</tr>
<tr>
<td>B</td>
<td>In all other cases.</td>
<td>METAPHONE ('Box') returns BKS.</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>X</td>
<td>When followed by IA or H.</td>
<td>METAPHONE ('Facial') returns FXL.</td>
</tr>
<tr>
<td>S</td>
<td>When followed by I, E, or Y.</td>
<td>METAPHONE ('Fence') returns FNS.</td>
<td></td>
</tr>
<tr>
<td>n/a</td>
<td>When it follows S, and is followed by I, E, or Y.</td>
<td>METAPHONE ('Scene') returns SN.</td>
<td></td>
</tr>
<tr>
<td>Input</td>
<td>Returns</td>
<td>Condition</td>
<td>Example</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------</td>
<td>------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>K</td>
<td>In all other cases.</td>
<td>METAPHONE ('Cool') returns KL.</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>J</td>
<td>When followed by GE, GY, or GI.</td>
<td>METAPHONE ('Dodge') returns TJ.</td>
</tr>
<tr>
<td>T</td>
<td>In all other cases.</td>
<td>METAPHONE ('David') returns TFT.</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>In all cases.</td>
<td>METAPHONE ('FOX') returns FKS.</td>
</tr>
<tr>
<td>G</td>
<td>F</td>
<td>When followed by H and the first character in the input string is not B, D, or H.</td>
<td>METAPHONE ('Tough') returns TF.</td>
</tr>
<tr>
<td>n/a</td>
<td>When followed by H and the first character in the input string is B, D, or H.</td>
<td>METAPHONE ('Hugh') returns HF.</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>When followed by I, E or Y and does not repeat.</td>
<td>METAPHONE ('Magic') returns MJK.</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>In all other cases.</td>
<td>METAPHONE ('GUN') returns KN.</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>When it does not follow C, G, P, S, or T and is followed by A, E, I, or U.</td>
<td>METAPHONE ('DHAT') returns THT.</td>
</tr>
<tr>
<td>n/a</td>
<td>In all other cases.</td>
<td>METAPHONE ('Chain') returns XN.</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>In all cases.</td>
<td>METAPHONE ('Jen') returns JN.</td>
</tr>
<tr>
<td>K</td>
<td>n/a</td>
<td>When it follows C.</td>
<td>METAPHONE ('Ckim') returns KM.</td>
</tr>
<tr>
<td>K</td>
<td>In all other cases.</td>
<td>METAPHONE ('Kim') returns KM.</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>In all cases.</td>
<td>METAPHONE ('Laura') returns LR.</td>
</tr>
<tr>
<td>M</td>
<td>M</td>
<td>In all cases.</td>
<td>METAPHONE ('Maggi') returns MK.</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>In all cases.</td>
<td>METAPHONE ('Nancy') returns NNS.</td>
</tr>
<tr>
<td>P</td>
<td>F</td>
<td>When followed by H.</td>
<td>METAPHONE ('Phone') returns FN.</td>
</tr>
<tr>
<td>Input</td>
<td>Returns</td>
<td>Condition</td>
<td>Example</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>P</td>
<td>In all other cases.</td>
<td>METAPHONE ('Pip') returns PP.</td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>K</td>
<td>In all cases.</td>
<td>METAPHONE ('Queen') returns KN.</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>In all cases.</td>
<td>METAPHONE ('Ray') returns R.</td>
</tr>
<tr>
<td>S</td>
<td>X</td>
<td>When followed by H, IO, IA, or CHW.</td>
<td>METAPHONE ('Cash') returns KX.</td>
</tr>
<tr>
<td>S</td>
<td>In all other cases.</td>
<td>METAPHONE ('Sing') returns SNK.</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>X</td>
<td>When followed by IA or IO.</td>
<td>METAPHONE ('Patio') returns PX.</td>
</tr>
<tr>
<td>0 *</td>
<td>When followed by H.</td>
<td>METAPHONE ('Thor') returns 0R.</td>
<td></td>
</tr>
<tr>
<td>n/a</td>
<td>When followed by CH.</td>
<td>METAPHONE ('Glitch') returns KLTX.</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>In all other cases.</td>
<td>METAPHONE ('Tim') returns TM.</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>F</td>
<td>In all cases.</td>
<td>METAPHONE ('Vin') returns FN.</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>When followed by A, E, I, O, or U.</td>
<td>METAPHONE ('Wang') returns WNK.</td>
</tr>
<tr>
<td>n/a</td>
<td>In all other cases.</td>
<td>METAPHONE ('When') returns HN.</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>KS</td>
<td>In all cases.</td>
<td>METAPHONE ('Six') returns SKS.</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>When followed by A, E, I, O, or U.</td>
<td>METAPHONE ('Yang') returns YNK.</td>
</tr>
<tr>
<td>n/a</td>
<td>In all other cases.</td>
<td>METAPHONE ('Bobby') returns BB.</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>S</td>
<td>In all cases.</td>
<td>METAPHONE ('Zack') returns SK.</td>
</tr>
</tbody>
</table>

* The integer 0.

- Skips the initial character and encodes the remaining string if the first two characters of the input string have one of the following values:
  - **KN.** For example, METAPHONE('KNOT') returns 'NT'.
  - **GN.** For example, METAPHONE('GNOB') returns 'NB'.
  - **PN.** For example, METAPHONE('PNRX') returns 'NRKS'.
  - **AE.** For example, METAPHONE('AERL') returns 'ERL'.

---

252 Appendix B: Functions
• If a character other than "C" occurs more than once in the input string, encodes the first occurrence only. For example, METAPHONE('BBOX') returns 'BKS' and METAPHONE('CCOX') returns 'KKKS'.

**Syntax**

```plaintext
METAPHONE( string [,length] )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>Must be a character string. Passes the value you want to encode. The first character must be a character in the English language alphabet (A-Z). You can enter any valid expression. Skips any non-alphabetic character in <code>string</code>.</td>
</tr>
<tr>
<td>length</td>
<td>Optional</td>
<td>Must be an integer greater than 0. Specifies the number of characters in <code>string</code> that you want to encode. You can enter any valid expression. When <code>length</code> is 0 or a value greater than the length of <code>string</code>, encodes the entire input string. Default is 0.</td>
</tr>
</tbody>
</table>

**Return Value**

String.

NULL if one of the following conditions is true:

• All values passed to the function are NULL.
• No character in `string` is a letter of the English alphabet.
• `string` is empty.

**Examples**

The following expression encodes the first two characters in `EMPLOYEE_NAME` column to a string:

```plaintext
METAPHONE( EMPLOYEE_NAME, 2 )
```

<table>
<thead>
<tr>
<th>Employee_Name</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>JH</td>
</tr>
<tr>
<td>*@$</td>
<td>NULL</td>
</tr>
<tr>
<td>P$%oc&amp;&amp;KMNL</td>
<td>PK</td>
</tr>
</tbody>
</table>

The following expression encodes the first four characters in `EMPLOYEE_NAME` column to a string:

```plaintext
METAPHONE( EMPLOYEE_NAME, 4 )
```

<table>
<thead>
<tr>
<th>Employee_Name</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>JHN</td>
</tr>
<tr>
<td>ABC</td>
<td>ABK</td>
</tr>
<tr>
<td>*@$</td>
<td>NULL</td>
</tr>
<tr>
<td>P$%oc&amp;&amp;KMNL</td>
<td>PKKM</td>
</tr>
</tbody>
</table>

**MOD**

Returns the remainder of a division calculation. For example, `MOD(8, 5)` returns 3.
Syntax

\[
\text{MOD( numeric_value, divisor )}
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. The values you want to divide. You can enter any valid expression.</td>
</tr>
<tr>
<td>divisor</td>
<td>Required</td>
<td>The numeric value you want to divide by. The divisor cannot be 0.</td>
</tr>
</tbody>
</table>

Return Value

Numeric value of the datatype you pass to the function. The remainder of the numeric value divided by the divisor.

NULL if a value passed to the function is NULL.

Example

The following expression returns the modulus of the values in the PRICE column divided by the values in the QTY column:

\[
\text{MOD( PRICE, QTY )}
\]

<table>
<thead>
<tr>
<th>PRICE</th>
<th>QTY</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.00</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>12.00</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>9.00</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>15.00</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>NULL</td>
<td>3</td>
<td>NULL</td>
</tr>
<tr>
<td>20.00</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>25.00</td>
<td>0</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The last row (25, 0) produced an error because you cannot divide by 0. To avoid dividing by 0, you can create an expression similar to the following, which returns the modulus of Price divided by Quantity only if the quantity is not 0. If the quantity is 0, the function returns NULL:

\[
\text{MOD( PRICE, IIF( QTY = 0, NULL, QTY ) )}
\]

<table>
<thead>
<tr>
<th>PRICE</th>
<th>QTY</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.00</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>12.00</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>9.00</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>15.00</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>NULL</td>
<td>3</td>
<td>NULL</td>
</tr>
<tr>
<td>20.00</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>25.00</td>
<td>0</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The last row (25, 0) produced a NULL rather than an error because the IIF function replaces NULL with the 0 in the QTY column.

\[\text{NPER}\]

Returns the number of periods for an investment based on a constant interest rate and periodic, constant payments.
Syntax

\[
\text{NPER}( \text{rate, present value, payment}[, \text{future value, type}] )
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rate</td>
<td>Required</td>
<td>Numeric. Interest rate earned in each period. Expressed as a decimal number. Divide the rate by 100 to express it as a decimal number. Must be greater than or equal to 0.</td>
</tr>
<tr>
<td>present value</td>
<td>Required</td>
<td>Numeric. Lump-sum amount a series of future payments is worth.</td>
</tr>
<tr>
<td>payment</td>
<td>Required</td>
<td>Numeric. Payment amount due per period. Must be a negative number.</td>
</tr>
<tr>
<td>future value</td>
<td>Optional</td>
<td>Numeric. Cash balance you want to attain after the last payment is made. If you omit this value, NPER uses 0.</td>
</tr>
<tr>
<td>type</td>
<td>Optional</td>
<td>Boolean. Timing of the payment. Enter 1 if payment is at the beginning of period. Enter 0 if payment is at the end of period. Default is 0. If you enter a value other than 0 or 1, Informatica Cloud treats the value as 1.</td>
</tr>
</tbody>
</table>

Return Value

Numeric.

Example

The present value of an investment is $2,000. Each payment is $500 and the future value of the investment is $20,000. The following expression returns 9 as the number of periods for which you need to make the payments:

\[
\text{NPER}( 0.01, -2000, -500, 20000, \text{TRUE} )
\]

Notes

To calculate interest rate earned in each period, divide the annual rate by the number of payments made in a year. For example, if you make monthly payments at an annual interest rate of 15 percent, the value of the Rate argument is 15% divided by 12. If you make annual payments, the value of the Rate argument is 15%.

The payment value and present value are negative because these are amounts that you pay.

PMT

Returns the payment for a loan based on constant payments and a constant interest rate.

Syntax

\[
\text{PMT}( \text{rate, terms, present value, future value, type} )
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rate</td>
<td>Required</td>
<td>Numeric. Interest rate of the loan for each period. Expressed as a decimal number. Divide the rate by 100 to express it as a decimal number. Must be greater than or equal to 0.</td>
</tr>
<tr>
<td>terms</td>
<td>Required</td>
<td>Numeric. Number of periods or payments. Must be greater than 0.</td>
</tr>
<tr>
<td>Argument</td>
<td>Required/Optional</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>present value</td>
<td>Required</td>
<td>Numeric. Principal for the loan.</td>
</tr>
<tr>
<td>future value</td>
<td>Optional</td>
<td>Numeric. Cash balance you want to attain after the last payment. If you omit this value, PMT uses 0.</td>
</tr>
<tr>
<td>type</td>
<td>Optional</td>
<td>Boolean. Timing of the payment. Enter 1 if payment is at the beginning of period. Enter 0 if payment is at the end of period. Default is 0. If you enter a value other than 0 or 1, Informatica Cloud treats the value as 1.</td>
</tr>
</tbody>
</table>

**Return Value**

Numeric.

**Example**

The following expression returns -2111.64 as the monthly payment amount of a loan:

```
PMT( 0.01, 10, 20000 )
```

**Notes**

To calculate interest rate earned in each period, divide the annual rate by the number of payments made in a year. For example, if you make monthly payments at an annual interest rate of 15%, the rate is 15%/12. If you make annual payments, the rate is 15%.

The payment value is negative because these are amounts that you pay.

---

**POWER**

Returns a value raised to the exponent you pass to the function.

**Syntax**

```
POWER( base, exponent )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>Required</td>
<td>Numeric value. This argument is the base value. You can enter any valid expression. If the base value is negative, the exponent must be an integer.</td>
</tr>
<tr>
<td>exponent</td>
<td>Required</td>
<td>Numeric value. This argument is the exponent value. You can enter any valid expression. If the base value is negative, the exponent must be an integer. In this case, the function rounds any decimal values to the nearest integer before returning a value.</td>
</tr>
</tbody>
</table>

**Return Value**

Double value.

NULL if you pass a null value to the function.
Example

The following expression returns the values in the Numbers column raised to the values in the Exponent column:

\[
\text{POWER( NUMBERS, EXPONENT )}
\]

<table>
<thead>
<tr>
<th>NUMBERS</th>
<th>EXPONENT</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>2.0</td>
<td>100</td>
</tr>
<tr>
<td>3.5</td>
<td>6.0</td>
<td>1838.265625</td>
</tr>
<tr>
<td>NULL</td>
<td>2.0</td>
<td>NULL</td>
</tr>
<tr>
<td>10.0</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>-3.0</td>
<td>-6.0</td>
<td>0.00137174211248285</td>
</tr>
<tr>
<td>3.0</td>
<td>-6.0</td>
<td>0.00137174211248285</td>
</tr>
<tr>
<td>-3.0</td>
<td>6.0</td>
<td>729.0</td>
</tr>
<tr>
<td>-3.0</td>
<td>5.5</td>
<td>729.0</td>
</tr>
</tbody>
</table>

The value -3.0 raised to 6 returns the same results as -3.0 raised to 5.5. If the base is negative, the exponent must be an integer. Otherwise, Informatica Cloud rounds the exponent to the nearest integer value.

PV

Returns the present value of an investment.

Syntax

\[
\text{PV( rate, terms, payment [, future value, type] )}
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rate</td>
<td>Required</td>
<td>Numeric. Interest rate earned in each period. Expressed as a decimal number. Divide the rate by 100 to express it as a decimal number. Must be greater than or equal to 0.</td>
</tr>
<tr>
<td>terms</td>
<td>Required</td>
<td>Numeric. Number of periods or payments. Must be greater than 0.</td>
</tr>
<tr>
<td>payment</td>
<td>Required</td>
<td>Numeric. Payment amount due per period. Must be a negative number.</td>
</tr>
<tr>
<td>future value</td>
<td>Optional</td>
<td>Numeric. Cash balance after the last payment. If you omit this value, PV uses 0.</td>
</tr>
<tr>
<td>type</td>
<td>Optional</td>
<td>Boolean. Timing of the payment. Enter 1 if payment is at the beginning of period. Enter 0 if payment is at the end of period. Default is 0. If you enter a value other than 0 or 1, Informatica Cloud treats the value as 1.</td>
</tr>
</tbody>
</table>

Return Value

Numeric.

Example

The following expression returns 12,524.43 as the amount you must deposit in the account today to have a future value of $20,000 in one year if you also deposit $500 at the beginning of each period:

\[
\text{PV( 0.0075, 12, -500, 20000, TRUE )}
\]
RAND

Returns a random number between 0 and 1. This is useful for probability calculations.

Syntax

```
RAND( seed )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>seed</td>
<td>Optional</td>
<td>Numeric. Starting value for Informatica Cloud to generate the random number. Value must be a constant. If you do not enter a seed, Informatica Cloud uses the current system time to derive the numbers of seconds since January 1, 1971. It uses this value as the seed.</td>
</tr>
</tbody>
</table>

Return Value

Numeric.

For the same seed, Informatica Cloud generates the same sequence of numbers.

Example

The following expression may return a value of 0.417022004702574:

```
RAND (1)
```

RATE

Returns the interest rate earned for each period by a security.

Syntax

```
RATE( terms, payment, present value[, future value, type] )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>terms</td>
<td>Required</td>
<td>Numeric. Number of periods or payments. Must be greater than 0.</td>
</tr>
<tr>
<td>payment</td>
<td>Required</td>
<td>Numeric. Payment amount due for each period. Must be a negative number.</td>
</tr>
<tr>
<td>present value</td>
<td>Required</td>
<td>Numeric. Lump-sum amount that a series of future payments is worth now.</td>
</tr>
<tr>
<td>future value</td>
<td>Optional</td>
<td>Numeric. Cash balance you want to attain after the last payment. For example, the future value of a loan is 0. If you omit this argument, RATE uses 0.</td>
</tr>
<tr>
<td>type</td>
<td>Optional</td>
<td>Boolean. Timing of the payment. Enter 1 if payment is at the beginning of period. Enter 0 if payment is at the end of period. Default is 0. If you enter a value other than 0 or 1, Informatica Cloud treats the value as 1.</td>
</tr>
</tbody>
</table>

Return Value

Numeric.
Example

The following expression returns 0.0077 as the monthly interest rate of a loan:

\[
\text{RATE}(48, -500, 20000)
\]

To calculate the annual interest rate of the loan, multiply 0.0077 by 12. The annual interest rate is 0.0924 or 9.24%.

REG_EXTRACT

Extracts subpatterns of a regular expression within an input value. For example, from a regular expression pattern for a full name, you can also extract the first name or last name.

Note: Use the REG_REPLACE function to replace a character pattern in a string with another character pattern.

Syntax

\[
\text{REG_EXTRACT}(\text{subject, pattern, subPatternNum})
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>Required</td>
<td>String datatype. Passes the value you want to compare against the regular expression pattern.</td>
</tr>
<tr>
<td>pattern</td>
<td>Required</td>
<td>String datatype. Regular expression pattern that you want to match. You must use perl compatible regular expression syntax. Enclose the pattern in single quotations.</td>
</tr>
<tr>
<td>subPatternNum</td>
<td>Optional</td>
<td>Integer value. Subpattern number of the regular expression you want to match. Use the following guidelines to determine the subpattern number:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1 or no value. Extracts the entire regular expression pattern.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2. Extracts the first regular expression subpattern.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 3. Extracts the second regular expression subpattern.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- n. Extracts the n-1 regular expression subpattern.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default is 1.</td>
</tr>
</tbody>
</table>

Using perl Compatible Regular Expression Syntax

You must use perl compatible regular expression syntax with REG_EXTRACT, REG_MATCH and REG_REPLACE functions.

The following table provides perl compatible regular expression syntax guidelines:

Table 14. perl Compatible Regular Expression Syntax Guidelines

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>Matches any one character.</td>
</tr>
<tr>
<td>[a-z]</td>
<td>Matches one instance of a letter. For example, [a-z][a-z] can match ab or CA.</td>
</tr>
<tr>
<td>\d</td>
<td>Matches one instance of any digit from 0-9.</td>
</tr>
<tr>
<td>\s</td>
<td>Matches a whitespace character.</td>
</tr>
</tbody>
</table>
### Syntax

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\w</code></td>
<td>Matches one alphanumeric character, including underscore <code>_</code></td>
</tr>
<tr>
<td><code>()</code></td>
<td>Groups an expression. For example, the parentheses in <code>(\d-\d-\d\d)</code> groups the expression <code>\d-\d</code>, which finds any two numbers followed by a hyphen and any two numbers, as in 12-34.</td>
</tr>
<tr>
<td><code>[]</code></td>
<td>Matches the number of characters. For example, <code>\d{3}</code> matches any three numbers, such as 650 or 510. Or, <code>[a-z]{2}</code> matches any two letters, such as CA or NY.</td>
</tr>
<tr>
<td><code>?</code></td>
<td>Matches the preceding character or group of characters zero or one time. For example, <code>\d{3}(\d{4})?</code> matches any three numbers, which can be followed by a hyphen and any four numbers.</td>
</tr>
<tr>
<td><code>*</code></td>
<td>(asterisk) Matches zero or more instances of the values that follow the asterisk. For example, <code>*0</code> is any value that precedes a 0.</td>
</tr>
<tr>
<td><code>+</code></td>
<td>Matches one or more instances of the values that follow the plus sign. For example, <code>\w+</code> is any value that follows an alphanumeric character.</td>
</tr>
</tbody>
</table>

For example, the following regular expression finds 5-digit U.S.A. zip codes, such as 93930, and 9-digit zip codes, such as 93930-5407:

```
\d{5}(\d{4})?
```

`\d{5}` refers to any five numbers, such as 93930. The parentheses surrounding `\d{4}` group this segment of the expression. The hyphen represents the hyphen of a 9-digit zip code, as in 93930-5407. `\d{4}` refers to any four numbers, such as 5407. The question mark states that the hyphen and last four digits are optional or can appear one time.

### Converting COBOL Syntax to perl Compatible Regular Expression Syntax

If you are familiar with COBOL syntax, you can use the following information to write perl compatible regular expressions.

The following table shows examples of COBOL syntax and their perl equivalents:

#### Table 15. COBOL Syntax and perl Compatible Regular Expression Syntax Compared

<table>
<thead>
<tr>
<th>COBOL Syntax</th>
<th>perl Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td><code>\d</code></td>
<td>Matches one instance of any digit from 0-9.</td>
</tr>
<tr>
<td>9999</td>
<td><code>\d\d\d\d</code> or <code>\d{4}</code></td>
<td>Matches any four digits from 0-9, as in 1234 or 5936.</td>
</tr>
<tr>
<td>x</td>
<td><code>[a-z]</code></td>
<td>Matches one instance of a letter.</td>
</tr>
<tr>
<td>9xx9</td>
<td><code>\d[a-z][a-z]\d</code></td>
<td>Matches any number followed by two letters and another number, as in 1ab2.</td>
</tr>
</tbody>
</table>

### Converting SQL Syntax to perl Compatible Regular Expression Syntax

If you are familiar with SQL syntax, you can use the following information to write perl compatible regular expressions.
The following table shows examples of SQL syntax and their perl equivalents:

Table 16. SQL Syntax and perl Compatible Regular Expression Syntax Compared

<table>
<thead>
<tr>
<th>SQL Syntax</th>
<th>perl Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>.*</td>
<td>Matches any string.</td>
</tr>
<tr>
<td>A%</td>
<td>A.*</td>
<td>Matches the letter 'A' followed by any string, as in Area.</td>
</tr>
<tr>
<td>_</td>
<td>.(a period)</td>
<td>Matches any one character.</td>
</tr>
<tr>
<td>A_</td>
<td>A</td>
<td>Matches &quot;A&quot; followed by any one character, such as AZ.</td>
</tr>
</tbody>
</table>

Return Value

Returns the value of the nth subpattern that is part of the input value. The nth subpattern is based on the value you specify for subPatternNum.

NULL if the input is a null value or if the pattern is null.

Example

You might use REG_EXTRACT in an expression to extract first names from a regular expression that matches first name and last name. For example, the following expression returns the first name of a regular expression:

```
REG_EXTRACT( Employee_Name, '((\w+)\s+(\w+))', 2 )
```

<table>
<thead>
<tr>
<th>Employee_Name</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>John</td>
</tr>
<tr>
<td>Alice Baer</td>
<td>Alice</td>
</tr>
<tr>
<td>Eric Chan</td>
<td>Eric</td>
</tr>
<tr>
<td>Bernice Davis</td>
<td>Bernice</td>
</tr>
<tr>
<td>Greg Brown</td>
<td>Greg</td>
</tr>
<tr>
<td>Steve Kim</td>
<td>Steve</td>
</tr>
</tbody>
</table>

REG_MATCH

Returns whether a value matches a regular expression pattern. This lets you validate data patterns, such as IDs, telephone numbers, postal codes, and state names.

Note: Use the REG_REPLACE function to replace a character pattern in a string with a new character pattern.

Syntax

```
REG_MATCH( subject, pattern )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>Required</td>
<td>String datatype. Passes the value you want to match against the regular expression pattern.</td>
</tr>
<tr>
<td>pattern</td>
<td>Required</td>
<td>String datatype. Regular expression pattern that you want to match. You must use perl compatible regular expression syntax. Enclose the pattern in single quotation marks. For more information, see &quot;REG_EXTRACT&quot; on page 259.</td>
</tr>
</tbody>
</table>
Return Value

TRUE if the data matches the pattern.
FALSE if the data does not match the pattern.
NULL if the input is a null value or if the pattern is NULL.

Example

You might use REG_MATCH in an expression to validate telephone numbers. For example, the following expression matches a 10-digit telephone number against the pattern and returns a Boolean value based on the match:

\[
\text{REG\_MATCH}\ (\text{Phone\_Number}, '\(\d\d\d-\d\d\d-\d\d\d\d\)' )
\]

<table>
<thead>
<tr>
<th>Phone_Number</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>408-555-1212</td>
<td>TRUE</td>
</tr>
<tr>
<td>510-555-1212</td>
<td>NULL</td>
</tr>
<tr>
<td>92 555 51212</td>
<td>FALSE</td>
</tr>
<tr>
<td>650-555-1212</td>
<td>TRUE</td>
</tr>
<tr>
<td>415-555-1212</td>
<td>TRUE</td>
</tr>
<tr>
<td>831 555 12123</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

Tip

You can also use REG_MATCH for the following tasks:

- To verify that a value matches a pattern. This use is similar to the SQL LIKE function.
- To verify that values are characters. This use is similar to the SQL IS_CHAR function.

To verify that a value matches a pattern, use a period (.) and an asterisk (*) with the REG_MATCH function in an expression. A period matches any one character. An asterisk matches 0 or more instances of values that follow it.

For example, use the following expression to find account numbers that begin with 1835:

\[
\text{REG\_MATCH}\ (\text{ACCOUNT\_NUMBER}, '\1835.**')
\]

To verify that values are characters, use a REG_MATCH function with the regular expression \([a-zA-Z]+\). a-z matches all lowercase characters. A-Z matches all uppercase characters. The plus sign (+) indicates that there should be at least one character.

For example, use the following expression to verify that a list of last names contain only characters:

\[
\text{REG\_MATCH}\ (\text{LAST\_NAME}, '\[a-zA-Z]+')
\]

REG_REPLACE

Replaces characters in a string with a another character pattern. By default, REG_REPLACE searches the input string for the character pattern you specify and replaces all occurrences with the replacement pattern. You can also indicate the number of occurrences of the pattern you want to replace in the string.
Syntax

\[
\text{REG\_REPLACE( subject, pattern, replace, numReplacements )}
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>Required</td>
<td>String datatype. Passes the string you want to search.</td>
</tr>
<tr>
<td>pattern</td>
<td>Required</td>
<td>String datatype. Passes the character string to be replaced. You must use perl compatible regular expression syntax. Enclose the pattern in single quotation marks. For more information, see “REG_EXTRACT” on page 259.</td>
</tr>
<tr>
<td>replace</td>
<td>Required</td>
<td>String datatype. Passes the new character string.</td>
</tr>
<tr>
<td>numReplacements</td>
<td>Optional</td>
<td>Numeric datatype. Specifies the number of occurrences you want to replace. If you omit this option, REG_REPLACE will replace all occurrences of the character string.</td>
</tr>
</tbody>
</table>

Return Value

String.

Example

The following expression removes additional spaces from the Employee name data for each row of the Employee\_name column:

\[
\text{REG\_REPLACE( Employee\_Name, '/s+', ' ')}
\]

<table>
<thead>
<tr>
<th>Employee_Name</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam Smith</td>
<td>Adam Smith</td>
</tr>
<tr>
<td>Greg Sanders</td>
<td>Greg Sanders</td>
</tr>
<tr>
<td>Sarah Fe</td>
<td>Sarah Fe</td>
</tr>
<tr>
<td>Sam Cooper</td>
<td>Sam Cooper</td>
</tr>
</tbody>
</table>

REPLACECHR

Replaces characters in a string with a single character or no character. REPLACECHR searches the input string for the characters you specify and replaces all occurrences of all characters with the new character you specify.

Syntax

\[
\text{REPLACECHR( CaseFlag, InputString, OldCharSet, NewChar )}
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CaseFlag</td>
<td>Required</td>
<td>Must be an integer. Determines whether the arguments in this function are case sensitive. You can enter any valid expression. When CaseFlag is a number other than 0, the function is case sensitive. When CaseFlag is a null value or 0, the function is not case sensitive.</td>
</tr>
<tr>
<td>InputString</td>
<td>Required</td>
<td>Must be a character string. Passes the string you want to search. You can enter any valid expression. If you pass a numeric value, the function converts it to a character string. If InputString is NULL, REPLACECHR returns NULL.</td>
</tr>
<tr>
<td>Argument</td>
<td>Required/Optional</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OldCharSet</td>
<td>Required</td>
<td>Must be a character string. The characters you want to replace. You can enter one or more characters. You can enter any valid expression. You can also enter a text literal enclosed within single quotation marks, for example, 'abc'. If you pass a numeric value, the function converts it to a character string. If OldCharSet is NULL or empty, REPLACECHR returns InputString.</td>
</tr>
<tr>
<td>NewChar</td>
<td>Required</td>
<td>Must be a character string. You can enter one character, an empty string, or NULL. You can enter any valid expression. If NewChar is NULL or empty, REPLACECHR removes all occurrences of all characters in OldCharSet in InputString. If NewChar contains more than one character, REPLACECHR uses the first character to replace OldCharSet.</td>
</tr>
</tbody>
</table>

**Return Value**

String.

Empty string if REPLACECHR removes all characters in InputString.

NULL if InputString is NULL.

InputString if OldCharSet is NULL or empty.

**Example**

The following expression removes the double quotation marks from web log data for each row in the WEBLOG column:

```
REPLACECHR( 0, WEBLOG, '"', NULL )
```

<table>
<thead>
<tr>
<th>WEBLOG</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;GET /news/index.html HTTP/1.1&quot;</td>
<td>GET /news/index.html HTTP/1.1</td>
</tr>
<tr>
<td>&quot;GET /companyinfo/index.html HTTP/1.1&quot;</td>
<td>GET /companyinfo/index.html HTTP/1.1</td>
</tr>
<tr>
<td>GET /companyinfo/index.html HTTP/1.1</td>
<td>GET /companyinfo/index.html HTTP/1.1</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expression removes multiple characters for each row in the WEBLOG column:

```
REPLACECHR ( 1, WEBLOG, ']['"', NULL )
```

<table>
<thead>
<tr>
<th>WEBLOG</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;GET /news/index.html HTTP/1.1&quot;</td>
<td>&quot;GET /news/index.html HTTP/1.1&quot;</td>
</tr>
<tr>
<td>&quot;GET /news/index.html HTTP/1.1&quot;</td>
<td>&quot;GET /news/index.html HTTP/1.1&quot;</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expression changes part of the value of the customer code for each row in the CUSTOMER_CODE column:

```
REPLACECHR ( 1, CUSTOMER_CODE, 'A', 'M' )
```

<table>
<thead>
<tr>
<th>CUSTOMER_CODE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABA</td>
<td>MBM</td>
</tr>
<tr>
<td>abA</td>
<td>abM</td>
</tr>
<tr>
<td>BBC</td>
<td>BBC</td>
</tr>
<tr>
<td>ACC</td>
<td>MCC</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>
The following expression changes part of the value of the customer code for each row in the CUSTOMER_CODE column:

```
REPLACECHR ( 0, CUSTOMER_CODE, 'A', 'M' )
```

<table>
<thead>
<tr>
<th>CUSTOMER_CODE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABA</td>
<td>MBM</td>
</tr>
<tr>
<td>abA</td>
<td>MbM</td>
</tr>
<tr>
<td>BBC</td>
<td>BBC</td>
</tr>
<tr>
<td>ACC</td>
<td>MCC</td>
</tr>
</tbody>
</table>

The following expression changes part of the value of the customer code for each row in the CUSTOMER_CODE column:

```
REPLACECHR ( 1, CUSTOMER_CODE, 'A', NULL )
```

<table>
<thead>
<tr>
<th>CUSTOMER_CODE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABA</td>
<td>B</td>
</tr>
<tr>
<td>BBC</td>
<td>BBC</td>
</tr>
<tr>
<td>ACC</td>
<td>CC</td>
</tr>
<tr>
<td>AAA</td>
<td>[empty string]</td>
</tr>
<tr>
<td>aaa</td>
<td>aaa</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expression removes multiple numbers for each row in the INPUT column:

```
REPLACECHR ( 1, INPUT, '14', NULL )
```

<table>
<thead>
<tr>
<th>INPUT</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345</td>
<td>235</td>
</tr>
<tr>
<td>4141</td>
<td>NULL</td>
</tr>
<tr>
<td>11111</td>
<td>5</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

When you want to use a single quotation mark (') in either OldCharSet or NewChar, you must use the CHR function. The single quotation mark is the only character that cannot be used inside a string literal.

The following expression removes multiple characters, including the single quotation mark, for each row in the INPUT column:

```
REPLACECHR ( 1, INPUT, CHR(39), NULL )
```

<table>
<thead>
<tr>
<th>INPUT</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Tom Smith' 'Laura Jones'</td>
<td>Tom Smith Laura Jones</td>
</tr>
<tr>
<td>'Tom's'</td>
<td>'Toms'</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

**REPLACESTR**

Replaces characters in a string with a single character, multiple characters, or no character. REPLACESTR searches the input string for all strings you specify and replaces them with the new string you specify.

**Syntax**

```
REPLACESTR ( CaseFlag, InputString, OldString1, [OldString2, ... OldStringN], NewString )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CaseFlag</td>
<td>Required</td>
<td>Must be an integer. Determines whether the arguments in this function are case sensitive. You can enter any valid expression. When CaseFlag is a number other than 0, the function is case sensitive.</td>
</tr>
<tr>
<td>Argument</td>
<td>Required/ Optional</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>CaseFlag</strong></td>
<td>Optional</td>
<td>When CaseFlag is a null value or 0, the function is not case sensitive.</td>
</tr>
<tr>
<td><strong>InputString</strong></td>
<td>Required</td>
<td>Must be a character string. Passes the strings you want to search. You can enter any valid expression. If you pass a numeric value, the function converts it to a character string. If InputString is NULL, REPLACESTR returns NULL.</td>
</tr>
<tr>
<td><strong>OldString</strong></td>
<td>Required</td>
<td>Must be a character string. The string you want to replace. You must enter at least one OldString argument. You can enter one or more characters per OldString argument. You can enter any valid expression. You can also enter a text literal enclosed within single quotation marks, for example, 'abc'. If you pass a numeric value, the function converts it to a character string. When REPLACESTR contains multiple OldString arguments, and one or more OldString arguments is NULL or empty, REPLACESTR ignores the OldString argument. When all OldString arguments are NULL or empty, REPLACESTR returns InputString. The function replaces the characters in the OldString arguments in the order they appear in the function. For example, if you enter multiple OldString arguments, the first OldString argument has precedence over the second OldString argument, and the second OldString argument has precedence over the third OldString argument. When REPLACESTR replaces a string, it places the cursor after the replaced characters in InputString before searching for the next match. For more information, see the examples.</td>
</tr>
<tr>
<td><strong>NewString</strong></td>
<td>Required</td>
<td>Must be a character string. You can enter one character, multiple characters, an empty string, or NULL. You can enter any valid expression. If NewString is NULL or empty, REPLACESTR removes all occurrences of OldString in InputString.</td>
</tr>
</tbody>
</table>

**Return Value**

String.

Empty string if REPLACESTR removes all characters in InputString.

NULL if InputString is NULL.

InputString if all OldString arguments are NULL or empty.

**Example**

The following expression removes double quotation marks and two different text strings from web log data for each row in the WEBLOG column:

```sql
REPLACESTR( 1, WEBLOG, '"', 'GET ', ' HTTP/1.1', NULL )
```

**WEBLOG**

```
"GET /news/index.html HTTP/1.1"
"GET /companyinfo/index.html HTTP/1.1"
GET /companyinfo/index.html
GET
NULL
```

**RETURN VALUE**

```
/news/index.html
/companyinfo/index.html
{empty string}
NULL
```

The following expression changes the title for certain values for each row in the TITLE column:

```sql
REPLACESTR( 1, TITLE, 'rs.', 'iss', 's.' )
```

**TITLE**

```
Mrs.
Miss
Mr.
MRS.
```

**RETURN VALUE**

```
Ms.
Ms.
Mr.
MRS.
```
The following expression changes the title for certain values for each row in the TITLE column:

```
REPLACESTR (0, TITLE, 'rs.', 'iss', 's.')
```

<table>
<thead>
<tr>
<th>TITLE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mrs.</td>
<td>Ms.</td>
</tr>
<tr>
<td>MRS.</td>
<td>Ms.</td>
</tr>
</tbody>
</table>

The following expression shows how the REPLACESTR function replaces multiple OldString arguments for each row in the INPUT column:

```
REPLACESTR (1, INPUT, 'ab', 'bc', '*')
```

<table>
<thead>
<tr>
<th>INPUT</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>abc</td>
<td>*c</td>
</tr>
<tr>
<td>abbc</td>
<td>**</td>
</tr>
<tr>
<td>abbbbc</td>
<td><em>bb</em></td>
</tr>
<tr>
<td>bc</td>
<td>*</td>
</tr>
</tbody>
</table>

The following expression shows how the REPLACESTR function replaces multiple OldString arguments for each row in the INPUT column:

```
REPLACESTR (1, INPUT, 'ab', 'bc', 'b')
```

<table>
<thead>
<tr>
<th>INPUT</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ab</td>
<td>b</td>
</tr>
<tr>
<td>bc</td>
<td>b</td>
</tr>
<tr>
<td>abc</td>
<td>bc</td>
</tr>
<tr>
<td>abbc</td>
<td>bb</td>
</tr>
<tr>
<td>abbbcc</td>
<td>bbc</td>
</tr>
</tbody>
</table>

When you want to use a single quotation mark (') in either OldString or NewString, you must use the CHR function. Use both the CHR and CONCAT functions to concatenate a single quotation mark onto a string. The single quotation mark is the only character that cannot be used inside a string literal. Consider the following example:

```
CONCAT( 'Joan', CONCAT(CHR(39), 's car'))
```

The return value is:

Joan's car

The following expression changes a string that includes the single quotation mark, for each row in the INPUT column:

```
REPLACESTR (1, INPUT, CONCAT('it', CONCAT(CHR(39), 's')), 'its')
```

<table>
<thead>
<tr>
<th>INPUT</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>it's</td>
<td>its</td>
</tr>
<tr>
<td>mit's</td>
<td>mits</td>
</tr>
<tr>
<td>mits</td>
<td>mits</td>
</tr>
<tr>
<td>mits'</td>
<td>mits'</td>
</tr>
</tbody>
</table>

**REVERSE**

Reverses the input string.

**Syntax**

```
REVERSE( string )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>Any character value. Value you want to reverse.</td>
</tr>
</tbody>
</table>
Return Value

String. Reverse of the input value.

Example

The following expression reverses the numbers of the customer code:

```
REVERSE(CUSTOMER_CODE)
```

<table>
<thead>
<tr>
<th>CUSTOMER_CODE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>1000</td>
</tr>
<tr>
<td>0002</td>
<td>2000</td>
</tr>
<tr>
<td>0003</td>
<td>3000</td>
</tr>
<tr>
<td>0004</td>
<td>4000</td>
</tr>
</tbody>
</table>

ROUND (Dates)

Rounds one part of a date. You can also use ROUND to round numbers.

This functions can round the following parts of a date:

- **Year.** Rounds the year portion of a date based on the month. If the month is between January and June, the function returns January 1 of the input year, and sets the time to 00:00:00. If the month is between July and December, the function returns January 1 of the next year with the time set to 00:00:00. For example, the expression `ROUND(06/30/1998 2:30:55, 'YY')` returns 01/01/1998 00:00:00, and `ROUND(07/1/1998 3:10:15, 'YY')` returns 01/01/1999 00:00:00.

- **Month.** Rounds the month portion of a date based on the day of the month. If the day of the month is between 1 and 15, it rounds the date to the first day of the input month with the time set to 00:00:00. If the day of the month is between 16 and the last day of the month, it rounds to the first day of the next month with the time set to 00:00:00. For example, the expression `ROUND(4/15/1998 12:15:00, 'MM')` returns 4/1/1998 00:00:00, and `ROUND(4/16/1998 8:24:19, 'MM')` returns 5/1/1998 00:00:00.

- **Day.** Rounds the day portion of the date based on the time. If the time is between 00:00:00 (12AM) and 11:59:59AM, the function returns the current date with the time set to 00:00:00 (12AM). If the time is 12:00:00 (12PM) or later, the function rounds the date to the next day with the time set to 00:00:00 (12AM). For example, the expression `ROUND(06/13/1998 2:30:45, 'DD')` returns 06/13/1998 00:00:00, and `ROUND(06/13/1998 22:30:45, 'DD')` returns 06/14/1998 00:00:00.

- **Hour.** Rounds the hour portion of the date based on the minutes in the hour. If the minute portion of the time is between 0 and 29, the function returns the current hour with the minutes and seconds set to 0. If the minute portion is 30 or greater, the function rounds to the next hour and sets the minutes and seconds to 0. For example, the expression `ROUND(04/01/1998 11:29:35, 'HH')` returns 04/01/1998 11:00:00, and `ROUND(04/01/1998 13:39:00, 'HH')` returns 04/01/1998 14:00:00.

- **Minute.** Rounds the minute portion of the date based on the seconds. If time has 0 to 29 seconds, the function returns the current minutes and sets the seconds to 0. If the time has 30 to 59 seconds, the function rounds to the next minute and sets the seconds to 0. For example, the expression `ROUND(05/22/1998 10:15:29, 'MI')` returns 05/22/1998 10:15:00, and `ROUND(05/22/1998 10:15:30, 'MI')` returns 05/22/1998 10:16:00.
Syntax

```
ROUND( date [, format ] )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>Required</td>
<td>Date/Time datatype. You can nest TO_DATE to convert strings to dates before rounding.</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>Enter a valid format string. This is the portion of the date that you want to round. You can round only one portion of the date. If you omit the format string, the function rounds the date to the nearest day.</td>
</tr>
</tbody>
</table>

Return Value

Date with the specified part rounded. ROUND returns a date in the same format as the source date. You can link the results of this function to any column with a Date/Time datatype.

NULL if you pass a null value to the function.

Example

The following expressions round the year portion of dates in the DATE_SHIPPED column:

```
ROUND( DATE_SHIPPED, 'Y' )
ROUND( DATE_SHIPPED, 'YY' )
ROUND( DATE_SHIPPED, 'YYY' )
ROUND( DATE_SHIPPED, 'YYYY' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 15 1998 2:10:30AM</td>
<td>Jan 1 1998 12:00:00AM</td>
</tr>
<tr>
<td>Apr 19 1998 1:31:20PM</td>
<td>Jan 1 1998 12:00:00AM</td>
</tr>
<tr>
<td>Dec 20 1998 3:29:55PM</td>
<td>Jan 1 1999 12:00:00AM</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions round the month portion of each date in the DATE_SHIPPED column:

```
ROUND( DATE_SHIPPED, 'MM' )
ROUND( DATE_SHIPPED, 'MON' )
ROUND( DATE_SHIPPED, 'MONTH' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 15 1998 2:10:30AM</td>
<td>Jan 1 1998 12:00:00AM</td>
</tr>
<tr>
<td>Apr 19 1998 1:31:20PM</td>
<td>May 1 1998 12:00:00AM</td>
</tr>
<tr>
<td>Dec 20 1998 3:29:55PM</td>
<td>Jan 1 1999 12:00:00AM</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions round the day portion of each date in the DATE_SHIPPED column:

```
ROUND( DATE_SHIPPED, 'D' )
ROUND( DATE_SHIPPED, 'DD' )
ROUND( DATE_SHIPPED, 'DDD' )
ROUND( DATE_SHIPPED, 'DY' )
ROUND( DATE_SHIPPED, 'DAY' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 15 1998 2:10:31AM</td>
<td>Jan 15 1998 12:00:00AM</td>
</tr>
<tr>
<td>Apr 19 1998 1:31:20PM</td>
<td>Apr 19 1998 12:00:00AM</td>
</tr>
<tr>
<td>Dec 20 1998 3:29:55PM</td>
<td>Dec 20 1998 12:00:00AM</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions round the hour portion of each date in the DATE_SHIPPED column:

```
ROUND( DATE_SHIPPED, 'HH' )
ROUND( DATE_SHIPPED, 'HH12' )
ROUND( DATE_SHIPPED, 'HH24' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 15 1998 2:10:31AM</td>
<td>Jan 15 1998 2:00:00AM</td>
</tr>
<tr>
<td>Apr 19 1998 1:31:20PM</td>
<td>Apr 19 1998 2:00:00PM</td>
</tr>
<tr>
<td>Dec 20 1998 3:29:55PM</td>
<td>Dec 20 1998 3:00:00PM</td>
</tr>
</tbody>
</table>
The following expression rounds the minute portion of each date in the DATE_SHIPPED column:

```sql
ROUND( DATE_SHIPPED, 'MI' )
```

### ROUND (Numbers)

Rounds numbers to a specified number of digits or decimal places. You can also use `ROUND` to round dates.

#### Syntax

```sql
ROUND( numeric_value [, precision] )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. You can enter any valid expression. Use operators to perform arithmetic before you round the values.</td>
</tr>
<tr>
<td>precision</td>
<td>Optional</td>
<td>Positive or negative integer. If you enter a positive precision, the function rounds to this number of decimal places. For example, ROUND(12.99, 1) returns 13.0 and ROUND(15.44, 1) returns 15.4. If you enter a negative precision, the function rounds this number of digits to the left of the decimal point, returning an integer. For example, ROUND(12.99, -1) returns 10 and ROUND(15.99, -1) returns 20. If you enter decimal precision, the function rounds to the nearest integer before evaluating the expression. For example, ROUND(12.99, 0.8) returns 13.0 because the function rounds 0.8 to 1 and then evaluates the expression. If you omit the precision argument, the function rounds to the nearest integer, truncating the decimal portion of the number. For example, ROUND(12.99) returns 13.</td>
</tr>
</tbody>
</table>

#### Return Value

Numeric value.

If one of the arguments is NULL, `ROUND` returns NULL.

#### Example

The following expression returns the values in the Price column rounded to three decimal places.

```sql
ROUND( PRICE, 3 )
```

<table>
<thead>
<tr>
<th>PRICE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.9936</td>
<td>12.994</td>
</tr>
<tr>
<td>15.9949</td>
<td>15.995</td>
</tr>
<tr>
<td>-18.8678</td>
<td>-18.868</td>
</tr>
<tr>
<td>56.9561</td>
<td>56.956</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>
You can round digits to the left of the decimal point by passing a negative integer in the `precision` argument:

```
ROUND( PRICE, -2 )
```

<table>
<thead>
<tr>
<th>PRICE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>13242.99</td>
<td>13200.0</td>
</tr>
<tr>
<td>1435.99</td>
<td>1400.0</td>
</tr>
<tr>
<td>-108.95</td>
<td>-100.0</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

If you pass a decimal value in the `precision` argument, Informatica Cloud rounds it to the nearest integer before evaluating the expression:

```
ROUND( PRICE, 0.8 )
```

<table>
<thead>
<tr>
<th>PRICE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.99</td>
<td>13.0</td>
</tr>
<tr>
<td>56.34</td>
<td>56.3</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

If you omit the `precision` argument, the function rounds to the nearest integer:

```
ROUND( PRICE )
```

<table>
<thead>
<tr>
<th>PRICE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.99</td>
<td>13.0</td>
</tr>
<tr>
<td>-15.99</td>
<td>-16.0</td>
</tr>
<tr>
<td>-18.99</td>
<td>-19.0</td>
</tr>
<tr>
<td>56.95</td>
<td>57.0</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

**Tip**

You can also use `ROUND` to explicitly set the precision of calculated values and achieve expected results.

### RPAD

Converts a string to a specified length by adding blanks or characters to the end of the string.

**Syntax**

```
RPAD( first_string, length [,second_string] )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>first_string</code></td>
<td>Required</td>
<td>Any string value. The strings you want to change. You can enter any valid expression.</td>
</tr>
<tr>
<td><code>length</code></td>
<td>Required</td>
<td>Must be a positive integer literal. Specifies the length you want each string to be. When <code>length</code> is a negative number, <code>RPAD</code> returns NULL.</td>
</tr>
<tr>
<td><code>second_string</code></td>
<td>Optional</td>
<td>Any string value. Passes the string you want to append to the right-side of the <code>first_string</code> values. Enclose the characters you want to add to the end of the string within single quotation marks, for example, 'abc'. This argument is case sensitive. If you omit the second string, the function pads the end of the first string with blanks.</td>
</tr>
</tbody>
</table>

**Return Value**

String of the specified length.

NULL if a value passed to the function is NULL or if length is a negative number.
Example

The following expression returns the item name with a length of 16 characters, appending the string ‘.’ to the end of each item name:

```
RPAD( ITEM_NAME, 16, '.')
```

<table>
<thead>
<tr>
<th>ITEM_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flashlight</td>
<td>Flashlight.......</td>
</tr>
<tr>
<td>Compass</td>
<td>Compass........</td>
</tr>
<tr>
<td>Regulator System</td>
<td>Regulator System</td>
</tr>
<tr>
<td>Safety Knife</td>
<td>Safety Knife.............</td>
</tr>
</tbody>
</table>

RPAD counts the length from left to right. So, if the first string is longer than the length, RPAD truncates the string from right to left. For example, RPAD(‘alphabetical’, 5, ‘x’) would return the string ‘alpha’. RPAD uses a partial part of the second_string when necessary.

The following expression returns the item name with a length of 16 characters, appending the string ‘*..*’ to the end of each item name:

```
RPAD( ITEM_NAME, 16, '*..*')
```

<table>
<thead>
<tr>
<th>ITEM_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flashlight</td>
<td>Flashlight*..**</td>
</tr>
<tr>
<td>Compass</td>
<td>Compass*..<strong>..</strong></td>
</tr>
<tr>
<td>Regulator System</td>
<td>Regulator System</td>
</tr>
<tr>
<td>Safety Knife</td>
<td>Safety Knife*..*</td>
</tr>
</tbody>
</table>

The following expression shows how RPAD handles negative values for the length argument for each row in the ITEM_NAME column:

```
RPAD( ITEM_NAME, -5, '.')
```

<table>
<thead>
<tr>
<th>ITEM_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flashlight</td>
<td>NULL</td>
</tr>
<tr>
<td>Compass</td>
<td>NULL</td>
</tr>
<tr>
<td>Regulator System</td>
<td>NULL</td>
</tr>
</tbody>
</table>

RTRIM

Removes blanks or characters from the end of a string.

If you do not specify a trim_set parameter in the expression, RTRIM removes only single-byte spaces.

If you use RTRIM to remove characters from a string, RTRIM compares the trim_set to each character in the string argument, character-by-character, starting with the right side of the string. If the character in the string matches any character in the trim_set, RTRIM removes it. RTRIM continues comparing and removing characters until it fails to find a matching character in the trim_set. It returns the string without the matching characters.
Syntax

\[ \text{RTRIM( string [, trim_set] )} \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>Any string. Passes the values you want to trim. You can enter any valid expression. Use operators to perform comparisons or concatenate strings before removing blanks from the end of a string.</td>
</tr>
<tr>
<td>trim_set</td>
<td>Optional</td>
<td>Any string value. Passes the characters you want to remove from the end of the string. You can also enter a text literal. However, you must enclose the characters you want to remove from the end of the string within single quotation marks, for example, ‘abc’. If you omit the second string, the function removes blanks from the end of the first string. RTRIM is case sensitive.</td>
</tr>
</tbody>
</table>

Return Value

String. The string values with the specified characters in the \textit{trim_set} argument removed.

NULL if a value passed to the function is NULL.

Example

The following expression removes the characters ‘re’ from the strings in the \textit{LAST\_NAME} column:

\[ \text{RTRIM( \textit{LAST\_NAME}, 're')} \]

<table>
<thead>
<tr>
<th>LAST_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nelson</td>
<td>Nelson</td>
</tr>
<tr>
<td>Page</td>
<td>Pag</td>
</tr>
<tr>
<td>Osborne</td>
<td>Osborn</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>Sawyer</td>
<td>Sawy</td>
</tr>
<tr>
<td>H. Bender</td>
<td>H. Bend</td>
</tr>
<tr>
<td>Steadman</td>
<td>Steadman</td>
</tr>
</tbody>
</table>

RTRIM removes ‘e’ from Page even though ‘r’ is the first character in the \textit{trim_set}. This is because RTRIM searches, character-by-character, for the set of characters you specify in the \textit{trim_set} argument. If the last character in the string matches the first character in the \textit{trim_set}, RTRIM removes it. If, however, the last character in the string does not match, RTRIM compares the second character in the \textit{trim_set}. If the second from last character in the string matches the second character in the \textit{trim_set}, RTRIM removes it, and so on. When the character in the string fails to match the \textit{trim_set}, RTRIM returns the string and evaluates the next row.

In the last example, the last character in Nelson does not match any character in the \textit{trim_set} argument, so RTRIM returns the string ‘Nelson’ and evaluates the next row.

Tips

Use RTRIM and LTRIM with \| or CONCAT to remove leading and trailing blanks after you concatenate two strings.

You can also remove multiple sets of characters by nesting RTRIM. For example, if you want to remove trailing blanks and the character ‘t’ from the end of each string in a column of names, you might create an expression similar to the following:

\[ \text{RTRIM( RTRIM( \textit{NAMES} ), 't' )} \]
SET_DATE_PART

Sets one part of a date/time value to a value you specify. With SET_DATE_PART, you can change the following parts of a date:

- **Year.** Change the year by entering a positive integer in the `value` argument. Use any of the year format strings: Y, YY, YYY, or YYYY to set the year. For example, the expression `SET_DATE_PART( SHIP_DATE, 'YY', 2001 )` changes the year to 2001 for all dates in the SHIP_DATE column.

- **Month.** Change the month by entering a positive integer between 1 and 12 (January=1 and December=12) in the `value` argument. Use any of the month format strings: MM, MON, MONTH to set the month. For example, the expression `SET_DATE_PART( SHIP_DATE, 'MONTH', 10 )` changes the month to October for all dates in the SHIP_DATE column.

- **Day.** Change the day by entering a positive integer between 1 and 31 (except for the months that have less than 31 days: February, April, June, September, and November) in the `value` argument. Use any of the month format strings (D, DD, DDD, DY, and DAY) to set the day. For example, the expression `SET_DATE_PART( SHIP_DATE, 'DD', 10 )` changes the day to 10 for all dates in the SHIP_DATE column.

- **Hour.** Change the hour by entering a positive integer between 0 and 24 (where 0=12AM, 12=12PM, and 24 =12AM) in the `value` argument. Use any of the hour format strings (HH, HH12, HH24) to set the hour. For example, the expression `SET_DATE_PART( SHIP_DATE, 'HH', 14 )` changes the hour to 14:00:00 (or 2:00:00PM) for all dates in the SHIP_DATE column.

- **Minute.** Change the minutes by entering a positive integer between 0 and 59 in the `value` argument. You use the MI format string to set the minute. For example, the expression `SET_DATE_PART( SHIP_DATE, 'MI', 25 )` changes the minute to 25 for all dates in the SHIP_DATE column.

- **Seconds.** You can change the seconds by entering a positive integer between 0 and 59 in the `value` argument. You use the SS format string to set the second. For example, the expression `SET_DATE_PART( SHIP_DATE, 'SS', 59 )` changes the second to 59 for all dates in the SHIP_DATE column.

**Syntax**

```
SET_DATE_PART( date, format, value )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>date</code></td>
<td>Required</td>
<td>Date/Time datatype. The date you want to modify. You can enter any valid expression.</td>
</tr>
<tr>
<td><code>format</code></td>
<td>Required</td>
<td>A format string specifying the portion of the date to be changed. The format string is not case sensitive.</td>
</tr>
<tr>
<td><code>value</code></td>
<td>Required</td>
<td>A positive integer value assigned to the specified portion of the date. The integer must be a valid value for the part of the date you want to change. If you enter an improper value (for example, February 30), the session fails.</td>
</tr>
</tbody>
</table>

**Return Value**

Date in the same format as the source date with the specified part changed.

NULL if a value passed to the function is NULL.
Example

The following expressions change the hour to 4PM for each date in the DATE_PROMISED column:

\[
\text{SET\_DATE\_PART}\left( \text{DATE\_PROMISED}, 'HH', 16 \right)
\]
\[
\text{SET\_DATE\_PART}\left( \text{DATE\_PROMISED}, 'HH12', 16 \right)
\]
\[
\text{SET\_DATE\_PART}\left( \text{DATE\_PROMISED}, 'HH24', 16 \right)
\]

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 1 1997 12:15:56AM</td>
<td>Jan 1 1997 4:15:56PM</td>
</tr>
<tr>
<td>Feb 13 1997 2:30:01AM</td>
<td>Feb 13 1997 4:30:01PM</td>
</tr>
<tr>
<td>Dec 12 1997 8:07:33AM</td>
<td>Dec 12 1997 4:07:33PM</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions change the month to June for the dates in the DATE_PROMISED column. Informatica Cloud displays an error when you try to create a date that does not exist, such as changing March 31 to June 31:

\[
\text{SET\_DATE\_PART}\left( \text{DATE\_PROMISED}, 'MM', 6 \right)
\]
\[
\text{SET\_DATE\_PART}\left( \text{DATE\_PROMISED}, 'MON', 6 \right)
\]
\[
\text{SET\_DATE\_PART}\left( \text{DATE\_PROMISED}, 'MONTH', 6 \right)
\]

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 13 1997 2:30:01AM</td>
<td>Jun 13 1997 2:30:01AM</td>
</tr>
<tr>
<td>Dec 12 1997 8:07:33AM</td>
<td>Jun 12 1997 8:07:33AM</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions change the year to 2000 for the dates in the DATE_PROMISED column:

\[
\text{SET\_DATE\_PART}\left( \text{DATE\_PROMISED}, 'Y', 2000 \right)
\]
\[
\text{SET\_DATE\_PART}\left( \text{DATE\_PROMISED}, 'YY', 2000 \right)
\]
\[
\text{SET\_DATE\_PART}\left( \text{DATE\_PROMISED}, 'YYY', 2000 \right)
\]
\[
\text{SET\_DATE\_PART}\left( \text{DATE\_PROMISED}, 'YYYY', 2000 \right)
\]

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 13 1997 2:30:01AM</td>
<td>Feb 13 2000 2:30:01AM</td>
</tr>
<tr>
<td>Dec 12 1997 8:07:33AM</td>
<td>Dec 12 2000 4:07:33PM</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Tip

If you want to change multiple parts of a date at one time, you can nest multiple SET_DATE_PART functions within the date argument. For example, you might write the following expression to change all of the dates in the DATE_ENTERED column to July 1 1998:

\[
\text{SET\_DATE\_PART}\left( \text{SET\_DATE\_PART}\left( \text{SET\_DATE\_PART}\left( \text{DATE\_ENTERED}, 'YYYY', 1998\right), 'MM', 7\right), 'DD', 1\right)
\]

SIGN

Returns whether a numeric value is positive, negative, or 0.

Syntax

\[
\text{SIGN}\left( \text{numeric\_value} \right)
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric value. Passes the values you want to evaluate. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

SIGN 275
Return Value

-1 for negative values.
0 for 0.
1 for positive values.
NULL if NULL.

Example

The following expression determines if the SALES column includes any negative values:

\[
\text{SIGN( SALES )}
\]

<table>
<thead>
<tr>
<th>Sales</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>-25.99</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

SIN

Returns the sine of a numeric value (expressed in radians).

Syntax

\[
\text{SIN( numeric\_value )}
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. Numeric data expressed in radians (degrees multiplied by ( \pi ) divided by 180). Passes the values for which you want to calculate the sine. You can enter any valid expression. You can also use operators to convert a numeric value to radians or perform arithmetic within the SIN calculation.</td>
</tr>
</tbody>
</table>

Return Value

Double value.

NULL if a value passed to the function is NULL.

Example

The following expression converts the values in the Degrees column to radians and then calculates the sine for each radian:

\[
\text{SIN( DEGREES * 3.14159265359 / 180 )}
\]

<table>
<thead>
<tr>
<th>Degrees</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>90</td>
<td>0.999847695156393</td>
</tr>
<tr>
<td>70</td>
<td>0.939692620785936</td>
</tr>
<tr>
<td>30</td>
<td>0.5000000000000003</td>
</tr>
<tr>
<td>5</td>
<td>0.087155742746639</td>
</tr>
<tr>
<td>18</td>
<td>0.309016994374967</td>
</tr>
<tr>
<td>89</td>
<td>0.999847695156393</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

You can perform arithmetic on the values passed to SIN before the function calculates the sine. For example:

\[
\text{SIN( ARCS * 3.14159265359 / 180 )}
\]
SINH

Returns the hyperbolic sine of the numeric value.

**Syntax**

```plaintext
SINH( numeric_value )
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. Numeric data expressed in radians (degrees multiplied by pi divided by 180). Passes the values for which you want to calculate the hyperbolic sine. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

**Return Value**

Double value.

NULL if a value passed to the function is NULL.

**Example**

The following expression returns the hyperbolic sine for the values in the ANGLES column:

```sql
SINH( ANGLES )
```

**Tip**

You can perform arithmetic on the values passed to SINH before the function calculates the hyperbolic sine. For example:

```sql
SINH( MEASURES.ARCS / 180 )
```

SOUNDEX

Encodes a string value into a four-character string.

SOUNDEX works for characters in the English alphabet (A-Z). It uses the first character of the input string as the first character in the return value and encodes the remaining three unique consonants as numbers.

SOUNDEX encodes characters according to the following list of rules:

- Uses the first character in `string` as the first character in the return value and encodes it in uppercase. For example, both SOUNDEX(‘John’) and SOUNDEX(‘john’) return ‘J500’.
- Encodes the first three unique consonants following the first character in `string` and ignores the rest. For example, both SOUNDEX(‘JohnRB’) and SOUNDEX(‘JohnRBCD’) return ‘J561’.
- Assigns a single code to consonants that sound alike.
The following table lists SOUNDEX encoding guidelines for consonants:

Table 17. SOUNDEX Encoding Guidelines for Consonants

<table>
<thead>
<tr>
<th>Code</th>
<th>Consonant</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B, P, F, V</td>
</tr>
<tr>
<td>2</td>
<td>C, S, G, J, K, Q, X, Z</td>
</tr>
<tr>
<td>3</td>
<td>D, T</td>
</tr>
<tr>
<td>4</td>
<td>L</td>
</tr>
<tr>
<td>5</td>
<td>M, N</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
</tr>
</tbody>
</table>

- Skips the characters A, E, I, O, U, H, and W unless one of them is the first character in string. For example, SOUNDEX(‘A123’) returns ‘A000’ and SOUNDEX(‘MAeiouhwC’) returns ‘M000’.
- If string produces fewer than four characters, SOUNDEX pads the resulting string with zeroes. For example, SOUNDEX(‘J’) returns ‘J000’.
- If string contains a set of consecutive consonants that use the same code listed in the table above, SOUNDEX encodes the first occurrence and skips the remaining occurrences in the set. For example, SOUNDEX(‘AbbdMN’) returns ‘A135’.
- Skips numbers in string. For example, both SOUNDEX(‘Joh12n’) and SOUNDEX(‘1John’) return ‘J500’.
- Returns NULL if string is NULL or if all the characters in string are not letters of the English alphabet.

**Syntax**

```
SOUNDEX( string )
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>Character string. Passes the string value you want to encode. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

**Return Value**

String.

NULL if one of the following conditions is true:

- If value passed to the function is NULL.
- No character in string is a letter of the English alphabet.
- string is empty.

**Example**

The following expression encodes the values in the EMPLOYEE_NAME column:

```
SOUNDEX( EMPLOYEE_NAME )
```

<table>
<thead>
<tr>
<th>EMPLOYEE_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>J500</td>
</tr>
<tr>
<td>William</td>
<td>W450</td>
</tr>
<tr>
<td>Jane</td>
<td>J500</td>
</tr>
</tbody>
</table>
### SQRT

Returns the square root of a non-negative numeric value.

**Syntax**

```sql
SQRT( numeric_value )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Positive numeric value. Passes the values for which you want to calculate a square root. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

**Return Value**

Double value.

NULL if a value passed to the function is NULL.

**Example**

The following expression returns the square root for the values in the NUMBERS column:

```sql
SQRT( NUMBERS )
```

<table>
<thead>
<tr>
<th>NUMBERS</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>-100</td>
<td>None. Informatica Cloud writes the row into the error log file.</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>60.54</td>
<td>7.78074546557076</td>
</tr>
</tbody>
</table>

The value -100 results in an error during the session, since the function SQRT only evaluates positive numeric values. If you pass a negative value or character value, Informatica Cloud writes the row into the error log file.

You can perform arithmetic on the values passed to SQRT before the function calculates the square root.

---

### SUBSTR

Returns a portion of a string. SUBSTR counts all characters, including blanks, starting at the beginning of the string.
Syntax

```sql
SUBSTR( string, start [,length] )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>Must be a character string. Passes the strings you want to search. You can enter any valid expression. If you pass a numeric value, the function converts it to a character string.</td>
</tr>
<tr>
<td>start</td>
<td>Required</td>
<td>Must be an integer. The position in the string where you want to start counting. You can enter any valid expression. If the start position is a positive number, SUBSTR locates the start position by counting from the beginning of the string. If the start position is a negative number, SUBSTR locates the start position by counting from the end of the string. If the start position is 0, SUBSTR searches from the first character in the string.</td>
</tr>
<tr>
<td>length</td>
<td>Optional</td>
<td>Must be an integer greater than 0. The number of characters you want SUBSTR to return. You can enter any valid expression. If you omit the length argument, SUBSTR returns all of the characters from the start position to the end of the string. If you pass a negative integer or 0, the function returns an empty string. If you pass a decimal, the function rounds it to the nearest integer value.</td>
</tr>
</tbody>
</table>

Return Value

String.

Empty string if you pass a negative or 0 length value.

NULL if a value passed to the function is NULL.

Example

The following expressions return the area code for each row in the PHONE column:

```sql
SUBSTR( PHONE, 0, 3 )
```

<table>
<thead>
<tr>
<th>PHONE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>809-555-0269</td>
<td>809</td>
</tr>
<tr>
<td>357-687-6708</td>
<td>357</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

```sql
SUBSTR( PHONE, 1, 3 )
```

<table>
<thead>
<tr>
<th>PHONE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>809-555-3915</td>
<td>809</td>
</tr>
<tr>
<td>357-687-6708</td>
<td>357</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions return the phone number without the area code for each row in the PHONE column:

```sql
SUBSTR( PHONE, 5, 8 )
```

<table>
<thead>
<tr>
<th>PHONE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>808-555-0269</td>
<td>555-0269</td>
</tr>
<tr>
<td>809-555-3915</td>
<td>555-3915</td>
</tr>
<tr>
<td>357-687-6708</td>
<td>687-6708</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

You can also pass a negative start value to return the phone number for each row in the PHONE column. The expression still reads the source string from left to right when returning the result of the length argument:

```sql
SUBSTR( PHONE, -8, 3 )
```

<table>
<thead>
<tr>
<th>PHONE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>808-555-0269</td>
<td>555</td>
</tr>
<tr>
<td>809-555-3915</td>
<td>555</td>
</tr>
<tr>
<td>357-687-6708</td>
<td>687</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>
You can nest INSTR in the start or length argument to search for a specific string and return its position.

The following expression evaluates a string, starting from the end of the string. The expression finds the last (rightmost) space in the string and then returns all characters preceding it:

```
SUBSTR(CUST_NAME, 1, INSTR(CUST_NAME, ' ',-1,1) - 1)
```

<table>
<thead>
<tr>
<th>CUST_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATRICIA JONES</td>
<td>PATRICIA</td>
</tr>
<tr>
<td>MARY ELLEN SHAH</td>
<td>MARY ELLEN</td>
</tr>
</tbody>
</table>

The following expression removes the character '#' from a string:

```
SUBSTR(CUST_ID, 1, INSTR(CUST_ID, '#') - 1) || SUBSTR(CUST_ID, INSTR(CUST_ID, '#') + 1)
```

When the length argument is longer than the string, SUBSTR returns all the characters from the start position to the end of the string. Consider the following example:

```
SUBSTR('abcd', 2, 8)
```

The return value is ‘bcd’. Compare this result to the following example:

```
SUBSTR('abcd', -2, 8)
```

The return value is ‘cd’.

**SYSTIMESTAMP**

Returns the current date and time with precision to the nanosecond of the system that hosts the Secure Agent that starts the task. The precision to which you can retrieve the date and time depends on the system that hosts the Secure Agent.

**Syntax**

```
SYSTIMESTAMP([format])
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>format</td>
<td>Optional</td>
<td>Precision to which you want to retrieve the timestamp. You can specify precision up to seconds (SS), milliseconds (MS), microseconds (US), or nanoseconds (NS). Enclose the format string within single quotation marks. The format string is not case sensitive. For example, to display the date and time to the precision of milliseconds use the following syntax: SYSTIMESTAMP('MS'). Default precision is microseconds (US).</td>
</tr>
</tbody>
</table>

**Return Value**

Timestamp. Returns date and time to the specified precision. Precision dependent on platform.

**Examples**

Your organization has an online order service and processes real-time data. You can use the SYSTIMESTAMP function to generate a primary key for each transaction in the target.

Create a task with the following field mappings:

<table>
<thead>
<tr>
<th>Source Column</th>
<th>Target Column</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer_Name</td>
<td>Customer_Name</td>
<td>n/a</td>
</tr>
<tr>
<td>Order_Qty</td>
<td>Order_Qty</td>
<td>n/a</td>
</tr>
<tr>
<td>Transaction_Id</td>
<td>SYSTIMESTAMP(SS)</td>
<td></td>
</tr>
</tbody>
</table>
At run time, the SYSTIMESTAMP generates the system time to the precision of microseconds for each row:

<table>
<thead>
<tr>
<th>Customer_Name</th>
<th>Order_Qty</th>
<th>Transaction_Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vani Deed</td>
<td>14</td>
<td>07/06/2007 18:00:30</td>
</tr>
<tr>
<td>Kalia Crop</td>
<td>3</td>
<td>07/06/2007 18:00:30</td>
</tr>
<tr>
<td>Vani Deed</td>
<td>6</td>
<td>07/06/2007 18:00:31</td>
</tr>
<tr>
<td>Harry Spoon</td>
<td>32</td>
<td>07/06/2007 18:00:31</td>
</tr>
</tbody>
</table>

**TAN**

Returns the tangent of a numeric value (expressed in radians).

**Syntax**

\[ \text{TAN}( \text{numeric\_value} ) \]

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. Numeric data expressed in radians (degrees multiplied by pi divided by 180). Passes the numeric values for which you want to calculate the tangent. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

**Return Value**

Double value.

NULL if a value passed to the function is NULL.

**Example**

The following expression returns the tangent for all values in the DEGREES column:

\[ \text{TAN}( \text{DEGREES} \times 3.14159 / 180 ) \]

<table>
<thead>
<tr>
<th>DEGREES</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>2.74747741945531</td>
</tr>
<tr>
<td>50</td>
<td>1.19175359259435</td>
</tr>
<tr>
<td>30</td>
<td>0.577350269189672</td>
</tr>
<tr>
<td>5</td>
<td>0.0874886635259298</td>
</tr>
<tr>
<td>18</td>
<td>0.324919696232929</td>
</tr>
<tr>
<td>89</td>
<td>57.2899616310952</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

**TANH**

Returns the hyperbolic tangent of the numeric value passed to this function.
Syntax

\[ \text{TANH( numeric\_value )} \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. Numeric data expressed in radians (degrees multiplied by pi divided by 180). Passes the numeric values for which you want to calculate the hyperbolic tangent. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

Double value.

NULL if a value passed to the function is NULL.

Example

The following expression returns the hyperbolic tangent for the values in the ANGLES column:

\[ \text{TANH( ANGLES )} \]

<table>
<thead>
<tr>
<th>ANGLES</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.761594155555765</td>
</tr>
<tr>
<td>2.897</td>
<td>0.993926947790665</td>
</tr>
<tr>
<td>3.66</td>
<td>0.99876551914886</td>
</tr>
<tr>
<td>5.45</td>
<td>0.999963084213409</td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.345</td>
<td>0.331933853503641</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Tip

You can perform arithmetic on the values passed to TANH before the function calculates the hyperbolic tangent. For example:

\[ \text{TANH( ARCS / 360 )} \]

TO\_BIGINT

Converts a string or numeric value to a bigint value. TO\_BIGINT syntax contains an optional argument that you can choose to round the number to the nearest integer or truncate the decimal portion. TO\_BIGINT ignores leading blanks.
Syntax

TO_BIGINT( value [, flag] )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>String or numeric datatype. Passes the value you want to convert to a bigint value. You can enter any valid expression.</td>
</tr>
<tr>
<td>flag</td>
<td>Optional</td>
<td>Specifies whether to truncate or round the decimal portion. The flag must be an integer literal or the constants TRUE or FALSE: - TO_BIGINT truncates the decimal portion when the flag is TRUE or a number other than 0. - TO_BIGINT rounds the value to the nearest integer if the flag is FALSE or 0 or if you omit this argument. The flag is not set by default.</td>
</tr>
</tbody>
</table>

Return Value

BigInt.

NULL if the value passed to the function is NULL.

0 if the value passed to the function contains alphanumeric characters.

Examples

The following expressions use values from the IN_TAX source column:

TO_BIGINT( IN_TAX, TRUE )

IN_TAX
'7,245,176,201,123,435.6789'
'7,245,176,201,123,435.2'
'7,245,176,201,123,435.2.48'
NULL
'A12.3Grove'
'176,201,123,435.87'
'7,245,176,201,123,435.2'
'7,245,176,201,123,435.23'
-9,223,372,036,854,775,806.9
9,223,372,036,854,775,806.9

RETURN VALUE
7,245,176,201,123,435
7,245,176,201,123,435
7,245,176,201,123,435
NULL
0
176,201,123,435
-7,245,176,201,123,435
-7,245,176,201,123,435
-9,223,372,036,854,775,806
9,223,372,036,854,775,806

TO_BIGINT( IN_TAX )

IN_TAX
'7,245,176,201,123,435.6789'
'7,245,176,201,123,435.2'
'7,245,176,201,123,435.348'
NULL
'A12.3Grove'
'176,201,123,435.87'
'7,245,176,201,123,435.6789'
'7,245,176,201,123,435.23'
-9,223,372,036,854,775,806.9
9,223,372,036,854,775,806.9

RETURN VALUE
7,245,176,201,123,436
7,245,176,201,123,435
7,245,176,201,123,435
NULL
0
176,201,123,436
-7,245,176,201,123,436
-7,245,176,201,123,435
-9,223,372,036,854,775,807
9,223,372,036,854,775,807

TO_CHAR (Dates)

Converts dates to character strings. TO_CHAR also converts numeric values to strings. You can convert the date into any format using the TO_CHAR format strings.
Syntax

\[
\text{TO_CHAR( } \text{date }, \text{format} )
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>Required</td>
<td>Date/Time datatype. Passes the date values you want to convert to character strings. You can enter any valid expression.</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>Enter a valid TO_CHAR format string. The format string defines the format of the return value, not the format for the values in the date argument. If you omit the format string, the function returns a string based on the default date format of MM/DD/YYYY HH24:MI:SS.</td>
</tr>
</tbody>
</table>

Return Value

String.

NULL if a value passed to the function is NULL.

Example

The following expression converts the dates in the DATE_PROMISED column to text in the format MON DD YYYY:

\[
\text{TO_CHAR( DATE_PROMISED, 'MON DD YYYY' )}
\]

If you omit the format_string argument, TO_CHAR returns a string in the default date format:

\[
\text{TO_CHAR( DATE_PROMISED )}
\]

The following expressions return the day of the week for each date in a column:

\[
\text{TO_CHAR( DATE_PROMISED, 'D' )}
\]

\[
\text{TO_CHAR( DATE_PROMISED, 'DAY' )}
\]

The following expression returns the day of the month for each date in a column:

\[
\text{TO_CHAR( DATE_PROMISED, 'DD' )}
\]
The following expression returns the day of the year for each date in a column:

\[
\text{TO\_CHAR( DATE\_PROMISED, 'DDD' )}
\]

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>04-01-1997 12:00:10AM</td>
<td>'091'</td>
</tr>
<tr>
<td>02-22-1997 01:31:10PM</td>
<td>'053'</td>
</tr>
<tr>
<td>10-24-1997 02:12:30PM</td>
<td>'297'</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions return the hour of the day for each date in a column:

\[
\text{TO\_CHAR( DATE\_PROMISED, 'HH')}
\]
\[
\text{TO\_CHAR( DATE\_PROMISED, 'HH12')}
\]

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>04-01-1997 12:00:10AM</td>
<td>'12'</td>
</tr>
<tr>
<td>02-22-1997 01:31:10PM</td>
<td>'01'</td>
</tr>
<tr>
<td>10-24-1997 02:12:30PM</td>
<td>'02'</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

\[
\text{TO\_CHAR( DATE\_PROMISED, 'HH24')}
\]

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>04-01-1997 12:00:10AM</td>
<td>'00'</td>
</tr>
<tr>
<td>02-22-1997 01:31:10PM</td>
<td>'13'</td>
</tr>
<tr>
<td>10-24-1997 11:12:30PM</td>
<td>'23'</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expression converts date values to MJD values expressed as strings:

\[
\text{TO\_CHAR( SHIP\_DATE, 'J')}
\]

<table>
<thead>
<tr>
<th>SHIP_DATE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dec 31 1999 03:59:59PM</td>
<td>2451544</td>
</tr>
<tr>
<td>Jan 1 1900 01:02:03AM</td>
<td>2415021</td>
</tr>
</tbody>
</table>

The following expression converts dates to strings in the format MM/DD/YY:

\[
\text{TO\_CHAR( SHIP\_DATE, 'MM/DD/RR')}
\]

<table>
<thead>
<tr>
<th>SHIP_DATE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/31/1999 01:02:03AM</td>
<td>12/31/99</td>
</tr>
<tr>
<td>09/15/1996 03:59:59PM</td>
<td>09/15/96</td>
</tr>
<tr>
<td>05/17/2003 12:13:14AM</td>
<td>05/17/03</td>
</tr>
</tbody>
</table>

You can also use the format string SSSSS in a TO\_CHAR expression. For example, the following expression
converts the dates in the SHIP\_DATE column to strings representing the total seconds since midnight:

\[
\text{TO\_CHAR( SHIP\_DATE, 'SSSSS')}
\]

<table>
<thead>
<tr>
<th>SHIP_DATE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/31/1999 01:02:03AM</td>
<td>3783</td>
</tr>
<tr>
<td>09/15/1996 03:59:59PM</td>
<td>86399</td>
</tr>
</tbody>
</table>

In TO\_CHAR expressions, the YY format string produces the same results as the RR format string.

The following expression converts dates to strings in the format MM/DD/YY:

\[
\text{TO\_CHAR( SHIP\_DATE, 'MM/DD/YY')}
\]

<table>
<thead>
<tr>
<th>SHIP_DATE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/31/1999 01:02:03AM</td>
<td>12/31/99</td>
</tr>
<tr>
<td>09/15/1996 03:59:59PM</td>
<td>09/15/96</td>
</tr>
<tr>
<td>05/17/2003 12:13:14AM</td>
<td>05/17/03</td>
</tr>
</tbody>
</table>

The following expression returns the week of the month for each date in a column:

\[
\text{TO\_CHAR( DATE\_PROMISED, 'W')}
\]

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>04-01-1997 12:00:10AM</td>
<td>'01'</td>
</tr>
<tr>
<td>02-22-1997 01:31:10PM</td>
<td>'04'</td>
</tr>
<tr>
<td>10-24-1997 02:12:30PM</td>
<td>'04'</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>
The following expression returns the week of the year for each date in a column:

```sql
TO_CHAR( DATE_PROMISED, 'WW' )
```

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>04-01-1997 12:00:10PM</td>
<td>'18'</td>
</tr>
<tr>
<td>02-22-1997 01:31:10AM</td>
<td>'08'</td>
</tr>
<tr>
<td>10-24-1997 02:12:30AM</td>
<td>'43'</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

**Tip**

You can combine `TO_CHAR` and `TO_DATE` to convert a numeric value for a month into the text value for a month using a function such as:

```sql
TO_CHAR( TO_DATE( numeric_month, 'MM' ), 'MONTH' )
```

---

**TO_CHAR (Numbers)**

Converts numeric values to text strings. `TO_CHAR` also converts dates to strings.

`TO_CHAR` converts numeric values to text strings as follows:

- Converts double values to strings of up to 16 digits and provides accuracy up to 15 digits. If you pass a number with more than 15 digits, `TO_CHAR` rounds the number to the sixteenth digit.
- Returns decimal notation for numbers in the ranges (-1e16,-1e-16] and [1e-16, 1e16). `TO_CHAR` returns scientific notation for numbers outside these ranges.

**Note:** Informatica Cloud converts the values 1e-16 and -1e16 to scientific notation, but returns the values 1e-16 and -1e-16 in decimal notation.

**Syntax**

```sql
TO_CHAR( numeric_value )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. The numeric value you want to convert to a string. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

**Return Value**

String.

NULL if a value passed to the function is NULL.

**Example**

The following expression converts the values in the `SALES` column to text:

```sql
TO_CHAR( SALES )
```

<table>
<thead>
<tr>
<th>SALES</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1010.99</td>
<td>'1010.99'</td>
</tr>
<tr>
<td>-15.62567</td>
<td>'-15.62567'</td>
</tr>
<tr>
<td>10842764968208837340</td>
<td>'*1.0842764968208837340e+019' (rounded to 16th digit)</td>
</tr>
<tr>
<td>1.234567890123456789e-10</td>
<td>'0.00000000001234567890123457' (greater than 1e-16 but less than 1e16)</td>
</tr>
<tr>
<td>1.23456789012345e17</td>
<td>'1.23456789012345e17' (greater than 1e16)</td>
</tr>
<tr>
<td>0</td>
<td>'0'</td>
</tr>
<tr>
<td>33.15</td>
<td>'33.15'</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>
TO_DATE

Converts a character string to a date datatype in the same format as the character string. You use the TO_DATE format strings to specify the format of the source strings.

The target column must be date/time for TO_DATE expressions.

If you are converting two-digit years with TO_DATE, use either the RR or YY format string. Do not use the YYYY format string.

Syntax

```
TO_DATE( string [, format] )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>Must be a string datatype. Passes the values that you want to convert to dates. You can enter any valid expression.</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>Enter a valid TO_DATE format string. The format string must match the parts of the string argument. For example, if you pass the string 'Mar 15 1998 12:43:10AM', you must use the format string 'MON DD YYYY HH12:MI:SSAM'. If you omit the format string, the string value must be in the default date of MM/DD/YYYY HH24:MI:SS.</td>
</tr>
</tbody>
</table>

Return Value

Date.

TO_DATE always returns a date and time. If you pass a string that does not have a time value, the date returned always includes the time 00:00:00. You can map the results of this function to any target column with a date datatype.

NULL if you pass a null value to this function.

Warning: The format of the TO_DATE string must match the format string including any date separators. If it does not, Informatica Cloud might return inaccurate values or skip the row.

Example

The following expression returns date values for the strings in the DATE_PROMISED column. TO_DATE always returns a date and time. If you pass a string that does not have a time value, the date returned always includes the time 00:00:00. If you run a session in the twentieth century, the century will be 19. The current year on the machine running Informatica Cloud is 1998:

```
TO_DATE( DATE_PROMISED, 'MM/DD/YY' )
```

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'01/22/98'</td>
<td>Jan 22 1998 00:00:00</td>
</tr>
<tr>
<td>'05/03/98'</td>
<td>May 3 1998 00:00:00</td>
</tr>
<tr>
<td>'11/10/98'</td>
<td>Nov 10 1998 00:00:00</td>
</tr>
<tr>
<td>'10/19/98'</td>
<td>Oct 19 1998 00:00:00</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expression returns date and time values for the strings in the DATE_PROMISED column. If you pass a string that does not have a time value, Informatica Cloud writes the row into the error log file. If you run a session in the twentieth century, the century will be 19. The current year on the machine running Informatica Cloud is 1998:

```
TO_DATE( DATE_PROMISED, 'MON DD YYYY HH12:MI:SSAM' )
```

<table>
<thead>
<tr>
<th>DATE_PROMISED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Jan 22 1998 02:14:56PM'</td>
<td>Jan 22 1998 02:14:56PM</td>
</tr>
</tbody>
</table>
The following expression converts strings in the SHIP_DATE_MJD_STRING column to date values in the default date format:

\[
\text{TO}\_\text{DATE}\ (\text{SHIP}\_\text{DATE}\_\text{MJD}\_\text{STR}, \ 'J')
\]

Because the J format string does not include the time portion of a date, the return values have the time set to 00:00:00.

The following expression converts a string to a four-digit year format. The current year is 1998:

\[
\text{TO}\_\text{DATE}( \text{DATE}\_\text{STR}, \ 'MM/DD/RR')
\]

Note: For the second row, RR returns the year 2005 and YY returns the year 1905.

The following expression converts a string to a four-digit year format. The current year is 1998:

\[
\text{TO}\_\text{DATE}( \text{DATE}\_\text{STR}, \ 'MM/DD/Y')
\]

The following expression converts strings that includes the seconds since midnight to date values:

\[
\text{TO}\_\text{DATE}( \text{DATE}\_\text{STR}, \ 'MM/DD/YYYY \ SSSSS')
\]

If the target accepts different date formats, use TO_DATE and IS_DATE with the DECODE function to test for acceptable formats. For example:

\[
\text{DECODE}( \text{TRUE}, \\
\begin{align*}
\text{--test first format} \\
\text{IS}\_\text{DATE}( \text{CLOSE}\_\text{DATE}, \ 'MM/DD/YYYY\ HH24:MI:SS' ), \\
\text{--if true, convert to date} \\
\text{TO}\_\text{DATE}( \text{CLOSE}\_\text{DATE}, \ 'MM/DD/YYYY\ HH24:MI:SS' ),
\end{align*}
\begin{align*}
\text{--test second format; if true, convert to date} \\
\text{IS}\_\text{DATE}( \text{CLOSE}\_\text{DATE}, \ 'MM/DD/YYYY' ), \text{TO}\_\text{DATE}( \text{CLOSE}\_\text{DATE}, \ 'MM/DD/YYYY' ),
\end{align*}
\begin{align*}
\text{--test third format; if true, convert to date} \\
\text{IS}\_\text{DATE}( \text{CLOSE}\_\text{DATE}, \ 'MON DD YYYY' ), \text{TO}\_\text{DATE}( \text{CLOSE}\_\text{DATE}, \ 'MON DD YYYY' ),
\end{align*}
\begin{align*}
\text{--if none of the above} \\
\text{ERROR}( \ 'NOT\ A\ VALID\ DATE' )
\end{align*}
\)
You can combine TO_CHAR and TO_DATE to convert a numeric value for a month into the text value for a month using a function such as:

\[
\text{TO_CHAR( TO_DATE( numeric\_month, 'MM' ), 'MONTH' )}
\]

### TO_DECIMAL

Converts a string or numeric value to a decimal value. TO_DECIMAL ignores leading spaces.

**Syntax**

\[
\text{TO\_DECIMAL( value [, scale] )}
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>Must be a string or numeric datatype. Passes the values you want to convert to decimals. You can enter any valid expression.</td>
</tr>
<tr>
<td>scale</td>
<td>Optional</td>
<td>Must be an integer literal between 0 and 28, inclusive. Specifies the number of digits allowed after the decimal point. If you omit this argument, the function returns a value with the same scale as the input value.</td>
</tr>
</tbody>
</table>

**Return Value**

Decimal of precision and scale between 0 and 28, inclusive.

0 if the value in the selected column is an empty string or a non-numeric character.

NULL if a value passed to the function is NULL.

**Example**

This expression uses values from the column IN\_TAX. The datatype is decimal with precision of 10 and scale of 3:

\[
\text{TO\_DECIMAL( IN\_TAX, 3 )}
\]

<table>
<thead>
<tr>
<th>IN_TAX</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'15.6789'</td>
<td>15.679</td>
</tr>
<tr>
<td>'60.2'</td>
<td>60.200</td>
</tr>
<tr>
<td>'118.348'</td>
<td>118.348</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>'A12.3Grove'</td>
<td>0</td>
</tr>
</tbody>
</table>

### TO_FLOAT

Converts a string or numeric value to a double-precision floating point number (the Double datatype). TO_FLOAT ignores leading spaces.
Syntax

TO_FLOAT( value )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>Must be a string or numeric datatype. Passes the values you want to convert to double values. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

Return Value

Double value.

0 if the value in the column is blank or a non-numeric character.

NULL if a value passed to this function is NULL.

Example

This expression uses values from the column IN_TAX:

```
TO_FLOAT( IN_TAX )
```

<table>
<thead>
<tr>
<th>IN_TAX</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'15.6789'</td>
<td>15.6789</td>
</tr>
<tr>
<td>'60.2'</td>
<td>60.2</td>
</tr>
<tr>
<td>'118.348'</td>
<td>118.348</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>'A12.3Grove'</td>
<td>0</td>
</tr>
</tbody>
</table>

TO_INTEGER

Converts a string or numeric value to an integer. TO_INTEGER syntax contains an optional argument that you can choose to round the number to the nearest integer or truncate the decimal portion. TO_INTEGER ignores leading spaces.

Syntax

TO_INTEGER( value [, flag] )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Required</td>
<td>String or numeric datatype. Passes the value you want to convert to an integer. You can enter any valid expression.</td>
</tr>
<tr>
<td>flag</td>
<td>Optional</td>
<td>Specifies whether to truncate or round the decimal portion. The flag must be an integer literal or the constants TRUE or FALSE:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TO_INTEGER truncates the decimal portion when the flag is TRUE or a number other than 0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- TO_INTEGER rounds the value to the nearest integer if the flag is FALSE or 0 or if you omit this argument.</td>
</tr>
</tbody>
</table>

Return Value

Integer.

NULL if the value passed to the function is NULL.
0 if the value passed to the function contains alphanumeric characters.

Example

The following expressions use values from the column IN_TAX:

TO_INTEGER( IN_TAX, TRUE )

<table>
<thead>
<tr>
<th>IN_TAX</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'15.6789'</td>
<td>15</td>
</tr>
<tr>
<td>'60.2'</td>
<td>60</td>
</tr>
<tr>
<td>'118.348'</td>
<td>118</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>'A12.3Grove'</td>
<td>0</td>
</tr>
<tr>
<td>' 123.87'</td>
<td>123</td>
</tr>
<tr>
<td>'-15.6789'</td>
<td>-15</td>
</tr>
<tr>
<td>'-15.23'</td>
<td>-15</td>
</tr>
</tbody>
</table>

TO_INTEGER( IN_TAX, FALSE )

<table>
<thead>
<tr>
<th>IN_TAX</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'15.6789'</td>
<td>16</td>
</tr>
<tr>
<td>'60.2'</td>
<td>60</td>
</tr>
<tr>
<td>'118.348'</td>
<td>118</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>'A12.3Grove'</td>
<td>0</td>
</tr>
<tr>
<td>' 123.87'</td>
<td>124</td>
</tr>
<tr>
<td>'-15.6789'</td>
<td>-16</td>
</tr>
<tr>
<td>'-15.23'</td>
<td>-15</td>
</tr>
</tbody>
</table>

TRUNC (Dates)

Truncates dates to a specific year, month, day, hour, or minute. You can also use TRUNC to truncate numbers.

You can truncate the following date parts:

- **Year.** If you truncate the year portion of the date, the function returns Jan 1 of the input year with the time set to 00:00:00. For example, the expression TRUNC(6/30/1997 2:30:55, 'YY') returns 1/1/1997 00:00:00, and TRUNC(12/1/1997 3:10:15, 'YY') returns 1/1/1997 00:00:00.

- **Month.** If you truncate the month portion of a date, the function returns the first day of the month with the time set to 00:00:00. For example, the expression TRUNC(4/15/1997 12:15:00, 'MM') returns 4/1/1997 00:00:00, and TRUNC(4/30/1997 3:15:46, 'MM') returns 4/1/1997 00:00:00.

- **Day.** If you truncate the day portion of a date, the function returns the date with the time set to 00:00:00. For example, the expression TRUNC(6/13/1997 2:30:45, 'DD') returns 6/13/1997 00:00:00, and TRUNC(12/13/1997 22:30:45, 'DD') returns 12/13/1997 00:00:00.

- **Hour.** If you truncate the hour portion of a date, the function returns the date with the minutes and seconds set to 0. For example, the expression TRUNC(4/1/1997 11:29:35, 'HH') returns 4/1/1997 11:00:00, and TRUNC(4/1/1997 13:39:00, 'HH') returns 4/1/1997 13:00:00.

- **Minute.** If you truncate the minute portion of a date, the function returns the date with the seconds set to 0. For example, the expression TRUNC(5/22/1997 10:15:29, 'MI') returns 5/22/1997 10:15:00, and TRUNC(5/22/1997 10:18:30, 'MI') returns 5/22/1997 10:18:00.
Syntax
TRUNC( date [, format] )

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>Required</td>
<td>Date/Time datatype. The date values you want to truncate. You can enter any valid expression that evaluates to a date.</td>
</tr>
<tr>
<td>format</td>
<td>Optional</td>
<td>Enter a valid format string. The format string is not case sensitive. If you omit the format string, the function truncates the time portion of the date, setting it to 00:00:00.</td>
</tr>
</tbody>
</table>

Return Value
Date.
NULL if a value passed to the function is NULL.

Example
The following expressions truncate the year portion of dates in the DATE_SHIPPED column:

```
TRUNC( DATE_SHIPPED, 'Y' )
TRUNC( DATE_SHIPPED, 'YY' )
TRUNC( DATE_SHIPPED, 'YYY' )
TRUNC( DATE_SHIPPED, 'YYYY' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 1998 2:10:30AM</td>
<td>Jan 1 1998 12:00:00AM</td>
</tr>
<tr>
<td>Apr 19 1998 1:31:20PM</td>
<td>Apr 1 1998 12:00:00AM</td>
</tr>
<tr>
<td>Jun 20 1998 3:50:04AM</td>
<td>Jun 1 1998 12:00:00AM</td>
</tr>
<tr>
<td>Dec 20 1998 3:29:55PM</td>
<td>Dec 1 1998 12:00:00AM</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions truncate the month portion of each date in the DATE_SHIPPED column:

```
TRUNC( DATE_SHIPPED, 'MM' )
TRUNC( DATE_SHIPPED, 'MON' )
TRUNC( DATE_SHIPPED, 'MONTH' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 1998 2:10:30AM</td>
<td>Jan 1 1998 12:00:00AM</td>
</tr>
<tr>
<td>Apr 19 1998 1:31:20PM</td>
<td>Apr 1 1998 12:00:00AM</td>
</tr>
<tr>
<td>Jun 20 1998 3:50:04AM</td>
<td>Jun 1 1998 12:00:00AM</td>
</tr>
<tr>
<td>Dec 20 1998 3:29:55PM</td>
<td>Dec 1 1998 12:00:00AM</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions truncate the day portion of each date in the DATE_SHIPPED column:

```
TRUNC( DATE_SHIPPED, 'D' )
TRUNC( DATE_SHIPPED, 'DD' )
TRUNC( DATE_SHIPPED, 'DDD' )
TRUNC( DATE_SHIPPED, 'DY' )
TRUNC( DATE_SHIPPED, 'DAY' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 15 1998 2:10:31AM</td>
<td>Jan 15 1998 12:00:00AM</td>
</tr>
<tr>
<td>Apr 19 1998 1:31:20PM</td>
<td>Apr 19 1998 12:00:00AM</td>
</tr>
<tr>
<td>Jun 20 1998 3:50:04AM</td>
<td>Jun 20 1998 12:00:00AM</td>
</tr>
<tr>
<td>Dec 20 1998 3:29:55PM</td>
<td>Dec 20 1998 12:00:00AM</td>
</tr>
<tr>
<td>Dec 31 1998 11:59:59PM</td>
<td>Dec 31 1998 12:00:00AM</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The following expressions truncate the hour portion of each date in the DATE_SHIPPED column:

```
TRUNC( DATE_SHIPPED, 'HH' )
TRUNC( DATE_SHIPPED, 'HH12' )
TRUNC( DATE_SHIPPED, 'HH24' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 15 1998 2:10:31AM</td>
<td>Jan 15 1998 2:00:00AM</td>
</tr>
<tr>
<td>Apr 19 1998 1:31:20PM</td>
<td>Apr 19 1998 1:00:00PM</td>
</tr>
<tr>
<td>Jun 20 1998 3:50:04AM</td>
<td>Jun 20 1998 3:00:00AM</td>
</tr>
<tr>
<td>Dec 20 1998 3:29:55PM</td>
<td>Dec 20 1998 3:00:00PM</td>
</tr>
<tr>
<td>Dec 31 1998 11:59:59PM</td>
<td>Dec 31 1998 11:00:00PM</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>
The following expression truncates the minute portion of each date in the DATE_SHIPPED column:

```
TRUNC( DATE_SHIPPED, 'MI' )
```

<table>
<thead>
<tr>
<th>DATE_SHIPPED</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apr 19 1998 1:31:00PM</td>
<td>Apr 19 1998 1:31:00PM</td>
</tr>
<tr>
<td>Jun 20 1998 3:50:00AM</td>
<td>Jun 20 1998 3:50:00AM</td>
</tr>
<tr>
<td>Dec 20 1998 3:29:00PM</td>
<td>Dec 20 1998 3:29:00PM</td>
</tr>
<tr>
<td>Dec 31 1998 11:59:00PM</td>
<td>Dec 31 1998 11:59:00PM</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

**TRUNC (Numbers)**

Truncates numbers to a specific digit. You can also use TRUNC to truncate dates.

**Syntax**

```
TRUNC( numeric_value [, precision] )
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric_value</td>
<td>Required</td>
<td>Numeric datatype. Passes the values you want to truncate. You can enter any valid expression that evaluates to a Numeric datatype.</td>
</tr>
<tr>
<td>precision</td>
<td>Optional</td>
<td>Can be a positive or negative integer. You can enter any valid expression that evaluates to an integer. The integer specifies the number of digits to truncate.</td>
</tr>
</tbody>
</table>

If `precision` is a positive integer, TRUNC returns `numeric_value` with the number of decimal places specified by `precision`. If `precision` is a negative integer, TRUNC changes the specified digits to the left of the decimal point to zeros. If you omit the `precision` argument, TRUNC truncates the decimal portion of `numeric_value` and returns an integer.

If you pass a decimal `precision` value, Informatica Cloud rounds `numeric_value` to the nearest integer before evaluating the expression.

**Return Value**

Numeric value.

NULL if one of the arguments is NULL.

**Example**

The following expressions truncate the values in the PRICE column:

```
TRUNC( PRICE, 3 )
```

<table>
<thead>
<tr>
<th>PRICE</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.9995</td>
<td>12.999</td>
</tr>
<tr>
<td>-18.8652</td>
<td>-18.865</td>
</tr>
<tr>
<td>56.9563</td>
<td>56.956</td>
</tr>
<tr>
<td>15.9928</td>
<td>15.992</td>
</tr>
</tbody>
</table>
### UPPER

Converts lowercase string characters to uppercase.

**Syntax**

\[ \text{UPPER( string )} \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Required</td>
<td>String datatype. Passes the values you want to change to uppercase text. You can enter any valid expression.</td>
</tr>
</tbody>
</table>

**Return Value**

Uppercase string. If the data contains multibyte characters, the return value depends on the code page of the Secure Agent that runs the task.

NULL if a value passed to the function is NULL.

**Example**

The following expression changes all names in the FIRST_NAME column to uppercase:

\[ \text{UPPER( FIRST_NAME )} \]

<table>
<thead>
<tr>
<th>FIRST_NAME</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ramona</td>
<td>RAMONA</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>THOMAS</td>
<td>THOMAS</td>
</tr>
<tr>
<td>PierreRe</td>
<td>PIERRE</td>
</tr>
<tr>
<td>Bernice</td>
<td>BERNICE</td>
</tr>
</tbody>
</table>
This appendix includes the following topics:

- Operator Precedence, 296
- Arithmetic Operators, 297
- String Operators, 298
- Comparison Operators, 298
- Logical Operators, 299

Operator Precedence

When you create an expression, you can use multiple operators and use operators within nested expressions. If you write an expression that includes multiple operators, Informatica Cloud evaluates the expression in the following order:

1. Arithmetic operators
2. String operators
3. Comparison operators
4. Logical operators

Informatica Cloud evaluates operators in the order they appear in the following table. It evaluates operators in an expression with equal precedence to all operators from left to right.

The following table lists the precedence for all transformation language operators:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( )</td>
<td>Parentheses.</td>
</tr>
<tr>
<td>+, -, NOT</td>
<td>Unary plus and minus and the logical NOT operator.</td>
</tr>
<tr>
<td>* , /, %</td>
<td>Multiplication, division, modulus.</td>
</tr>
<tr>
<td>+, -</td>
<td>Addition, subtraction.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;, &lt;=, &gt;, &gt;=</td>
<td>Less than, less than or equal to, greater than, greater than or equal to.</td>
</tr>
</tbody>
</table>
## Arithmetic Operators

Use arithmetic operators to perform mathematical calculations on numeric data.

The following table lists the transformation language arithmetic operators in order of precedence:

### Table 18. Arithmetic Operators in the Transformation Language

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>+, -</td>
<td>Unary plus and minus. Unary plus indicates a positive value. Unary minus indicates a negative value.</td>
</tr>
<tr>
<td>*, /, %</td>
<td>Multiplication, division, modulus. A modulus is the remainder after dividing two integers. For example, 13 % 2 = 1 because 13 divided by 2 equals 6 with a remainder of 1.</td>
</tr>
<tr>
<td>+, -</td>
<td>Addition, subtraction. The addition operator (+) does not concatenate strings. To concatenate strings, use the string operator</td>
</tr>
</tbody>
</table>

If you perform arithmetic on a null value, the function returns NULL.

When you use arithmetic operators in an expression, all of the operands in the expression must be numeric. For example, the expression `1 + '1'` is not valid because it adds an integer to a string. The expression `1.23 + 4 / 2` is valid because all of the operands are numeric.
String Operators

Use the || string operator to concatenate two strings. The || operator converts operands of any datatype (except Binary) to String datatypes before concatenation:

<table>
<thead>
<tr>
<th>Input Value</th>
<th>Return Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>'alpha'</td>
<td></td>
</tr>
<tr>
<td>'alpha'</td>
<td></td>
</tr>
<tr>
<td>'alpha'</td>
<td></td>
</tr>
</tbody>
</table>

The || operator includes leading and trailing spaces. Use the LTRIM and RTRIM functions to trim leading and trailing spaces before concatenating two strings.

Nulls

The || operator ignores null values. However, if both values are NULL, the || operator returns NULL.

Example of String Operators

The following example shows an expression that concatenates employee first names and employee last names from two columns. This expression removes the spaces from the end of the first name and the beginning of the last name, concatenates a space to the end of each first name, then concatenates the last name:

LTRIM( RTRIM( EMP_FIRST ) || ' ' || LTRIM( EMP_LAST ))

<table>
<thead>
<tr>
<th>EMP_FIRST</th>
<th>EMP_LAST</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>' Alfred'</td>
<td>' Rice '</td>
<td>Alfred Rice</td>
</tr>
<tr>
<td>' Bernice'</td>
<td>' Kersins'</td>
<td>Bernice Kersins</td>
</tr>
<tr>
<td>NULL</td>
<td>' Proud'</td>
<td>Proud</td>
</tr>
<tr>
<td>' Curt'</td>
<td>NULL</td>
<td>Curt</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Note: You can also use the CONCAT function to concatenate two string values. The || operator, however, produces the same results in less time.

Comparison Operators

Use comparison operators to compare character or numeric strings, manipulate data, and return a TRUE (1) or FALSE (0) value.

The following table lists the transformation language comparison operators:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equal to.</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than.</td>
</tr>
</tbody>
</table>
Use the greater than (>) and less than (<) operators to compare numeric values or return a range of rows based on the sort order for a primary key in a particular field.

When you use comparison operators in an expression, the operands must be the same datatype. For example, the expression `123.4 > '123'` is not valid because the expression compares a decimal with a string. The expressions `123.4 > 123` and `'a' != 'b'` are valid because the operands are the same datatype.

If you compare a value to a null value, the result is NULL.

If a filter condition evaluates to NULL, Informatica Cloud returns NULL.

### Logical Operators

Use logical operators to manipulate numeric data. Expressions that return a numeric value evaluate to TRUE for values other than 0, FALSE for 0, and NULL for NULL.

The following table lists the transformation language logical operators:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NOT</strong></td>
<td>Negates result of an expression. For example, if an expression evaluates to TRUE, the operator NOT returns FALSE. If an expression evaluates to FALSE, NOT returns TRUE.</td>
</tr>
<tr>
<td><strong>AND</strong></td>
<td>Joins two conditions and returns TRUE if both conditions evaluate to TRUE. Returns FALSE if one condition is not true.</td>
</tr>
<tr>
<td><strong>OR</strong></td>
<td>Connects two conditions and returns TRUE if any condition evaluates to TRUE. Returns FALSE if both conditions are not true.</td>
</tr>
</tbody>
</table>

### Nulls

Expressions that combine a null value with a Boolean expression produce results that are ANSI compliant. For example, Informatica Cloud produces the following results:

- NULL AND TRUE = NULL
- NULL AND FALSE = FALSE
Appendix D

Constants

This appendix includes the following topics:
- FALSE, 300
- NULL, 300
- TRUE, 301

FALSE

Clarifies a conditional expression. FALSE is equivalent to the integer 0.

Example

The following example uses FALSE in a DECODE expression to return values based on the results of a comparison. This is useful if you want to perform multiple searches based on a single search value:

```sql
DECODE ( FALSE,
    Var1 = 22, 'Variable 1 was 22!',
    Var2 = 49, 'Variable 2 was 49!,'
    Var1 < 23, 'Variable 1 was less than 23.',
    Var2 > 30, 'Variable 2 was more than 30.',
    'Variables were out of desired ranges.' )
```

NULL

Indicates that a value is either unknown or undefined. NULL is not equivalent to a blank or empty string (for character columns) or 0 (for numerical columns).

Although you can write expressions that return nulls, any column that has the NOT NULL or PRIMARY KEY constraint will not accept nulls. Therefore, if Informatica Cloud tries to write a null value to a column with one of these constraints, the database will reject the row and Informatica Cloud will write it to the reject file. Be sure to consider nulls when you create transformations.

Functions can handle nulls differently. If you pass a null value to a function, it might return 0 or NULL, or it might ignore null values.
Working with Null Values in Boolean Expressions

Expressions that combine a null value with a Boolean expression produces results that are ANSI compliant. For example:

- NULL AND TRUE = NULL
- NULL AND FALSE = FALSE

Null Values in Filter Conditions

If a filter condition evaluates to NULL, the function does not select the record.

Nulls with Operators

Any expression that uses operators (except the string operator ||) and contains a null value always evaluates to NULL. For example, the following expression evaluates to NULL:

8 * 10 - NULL

To test for nulls, use the ISNULL function.

TRUE

Returns a value based on the result of a comparison. TRUE is equivalent to the integer 1.

Example

The following example uses TRUE in a DECODE expression to return values based on the results of a comparison. This is useful if you want to perform multiple searches based on a single search value:

```
DECODE( TRUE,
    Var1 = 22, 'Variable 1 was 22!,'
    Var2 = 49, 'Variable 2 was 49!,'
    Var1 < 23, 'Variable 1 was less than 23.,'
    Var2 > 30, 'Variable 2 was more than 30.',
    'Variables were out of desired ranges.' )
```
Dates

This appendix includes the following topics:

- Dates Overview, 302
- Date Format Strings, 305
- TO_CHAR Format Strings, 306
- TO_DATE and IS_DATE Format Strings, 309
- Understanding Date Arithmetic, 312

Dates Overview

The appendix provides information about using date functions and built-in date variables to perform transformations on dates.

With the date functions, you can round, truncate, or compare dates, extract one part of a date, or perform arithmetic on a date. You can pass any value with a date datatype to a date function.

Use date variables to capture the current date or session start time on the machine hosting Informatica Cloud.

The Informatica Cloud transformation language also provides the following sets of format strings:

- **Date format strings.** Use with date functions to specify the parts of a date.
- **TO_CHAR format strings.** Use to specify the format of the return string.
- **TO_DATE and IS_DATE format strings.** Use to specify the format of a string you want to convert to a date or test.

Date/Time Datatype

The transformation language provides a set of generic datatypes to transform data from different sources. These transformation datatypes include a Date/Time datatype. Informatica Cloud stores dates internally in binary format.

Date functions accept datetime values only. To pass a string to a date function, first use TO_DATE to convert it to a datetime value. For example, the following expression converts a string field to datetime values and then adds one month to each date:

\[
\text{ADD_TO_DATE( TO_DATE( STRING_PORT, 'MM/DD/RR'), 'MM', 1 })
\]

**Note:** Informatica Cloud supports dates between 1753 A.D. and 9999 A.D.
Milliseconds
Informatica Cloud supports datetime values up to the second. If you import a datetime value that includes milliseconds, Informatica Cloud truncates to seconds. If you write a datetime value to a target column that supports milliseconds, Informatica Cloud inserts zeros for the millisecond portion of the date.

Julian Day, Modified Julian Day, and the Gregorian Calendar
Informatica Cloud supports dates in the Gregorian calendar system only. Dates expressed in a different calendar system are not supported.

Note: Dates in the Julian calendar are called Julian dates and are not supported in Informatica Cloud. This term should not be confused with Julian Day or with Modified Julian Day.

The transformation language provides the ability to manipulate Modified Julian Day (MJD) formats using the J format string.

The MJD for a given date is the number of days to that date since Jan 1 4713 BC 00:00:00 (midnight). By definition, MJD includes a time component expressed as a decimal, which represents some fraction of 24 hours. The J format string does not convert this time component.

For example, the following TO_DATE expression converts strings in the SHIP_DATE_MJD_STRING field to date values in the default date format:

```
TO_DATE (SHIP_DATE_MJD_STR, 'J')
```

<table>
<thead>
<tr>
<th>SHIP_DATE_MJD_STR</th>
<th>RETURN_VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2451544</td>
<td>Dec 31 1999 00:00:00</td>
</tr>
<tr>
<td>2415021</td>
<td>Jan 1 1900 00:00:00</td>
</tr>
</tbody>
</table>

Because the J format string does not include the time portion of a date, the return values have the time set to 00:00:00.

You can also use the J format string in TO_CHAR expressions. For example, use the J format string in a TO_CHAR expression to convert date values to MJD values expressed as strings. For example:

```
TO_CHAR(SHIP_DATE, 'J')
```

<table>
<thead>
<tr>
<th>SHIP_DATE</th>
<th>RETURN_VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dec 31 1999 23:59:59</td>
<td>2451544</td>
</tr>
<tr>
<td>Jan 1 1900 01:02:03</td>
<td>2415021</td>
</tr>
</tbody>
</table>

Note: Informatica Cloud ignores the time portion of the date in a TO_CHAR expression.

Dates in the Year 2000
All transformation language date functions support the year 2000. Informatica Cloud supports dates between 1753 A.D. and 9999 A.D.

RR Format String
The transformation language provides the RR format string to convert strings with two-digit years to dates. Using TO_DATE and the RR format string, you can convert a string in the format MM/DD/RR to a date. The RR format string converts data differently depending on the current year.

- **Current Year Between 0 and 49.** If the current year is between 0 and 49 (such as 2003) and the source string year is between 0 and 49, Informatica Cloud returns the current century plus the two-digit year from the source string. If the source string year is between 50 and 99, Informatica Cloud returns the previous century plus the two-digit year from the source string.
• **Current Year Between 50 and 99.** If the current year is between 50 and 99 (such as 1998) and the source string year is between 0 and 49, Informatica Cloud returns the next century plus the two-digit year from the source string. If the source string year is between 50 and 99, Informatica Cloud returns the current century plus the specified two-digit year.

The following table summarizes how the RR format string converts to dates:

**Table 21. RR Format String Conversions**

<table>
<thead>
<tr>
<th>Current year</th>
<th>Source year</th>
<th>RR Format String Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-49</td>
<td>0-49</td>
<td>Current century</td>
</tr>
<tr>
<td>0-49</td>
<td>50-99</td>
<td>Previous century</td>
</tr>
<tr>
<td>50-99</td>
<td>0-49</td>
<td>Next century</td>
</tr>
<tr>
<td>50-99</td>
<td>50-99</td>
<td>Current century</td>
</tr>
</tbody>
</table>

**Example of RR**

The following expression produces the same return values for any current year between 1950 and 2049:

```sql
TO_DATE( ORDER_DATE, 'MM/DD/RR' )
```

<table>
<thead>
<tr>
<th>ORDER_DATE</th>
<th>RETURN_VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>'04/12/98'</td>
<td>04/12/1998 00:00:00</td>
</tr>
<tr>
<td>'11/09/01'</td>
<td>11/09/2001 00:00:00</td>
</tr>
</tbody>
</table>

**Difference Between the YY and RR Format Strings**

The transformation language also provides a YY format string. Both the RR and YY format strings specify two-digit years. The YY and RR format strings produce identical results when used with all date functions except TO_DATE. In TO_DATE expressions, RR and YY produce different results.

The following table shows the different results each format string returns:

**Table 22. Differences Between RR and YY Format Strings**

<table>
<thead>
<tr>
<th>String</th>
<th>Current Year</th>
<th>TO_DATE(String, 'MM/DD/RR')</th>
<th>TO_DATE(String, 'MM/DD/YY')</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/12/98</td>
<td>1998</td>
<td>04/12/1998 00:00:00</td>
<td>04/12/1998 00:00:00</td>
</tr>
<tr>
<td>11/09/01</td>
<td>1998</td>
<td>11/09/2001 00:00:00</td>
<td>11/09/1901 00:00:00</td>
</tr>
<tr>
<td>04/12/98</td>
<td>2003</td>
<td>04/12/1998 00:00:00</td>
<td>04/12/2098 00:00:00</td>
</tr>
<tr>
<td>11/09/01</td>
<td>2003</td>
<td>11/09/2001 00:00:00</td>
<td>11/09/2001 00:00:00</td>
</tr>
</tbody>
</table>

For dates in the year 2000 and beyond, the YY format string produces less meaningful results than the RR format string. Use the RR format string for dates in the twenty-first century.

**Dates in Databases**

In general, dates stored in databases contain a date and time value. The date includes the month, day, and year, while the time might include the hours, minutes, and seconds. You can pass datetime data to any of the date

---

304 Appendix E: Dates
functions. Although date formats vary from database to database, and even between applications, Informatica Cloud can read any date with a date datatype.

**Dates in Flat Files**

The transformation language provides the TO_DATE function to convert strings to datetime values. You can also use IS_DATE to check if a string is a valid date before converting it with TO_DATE.

**Note:** Transformation language date functions accept date values only. If you want to pass a string to a date function, you must first use the TO_DATE function to convert it to a transformation Date/Time datatype.

**Default Date Format**

The service uses a default date format to store and manipulate strings that represent dates. Because Informatica Cloud stores dates in binary format, Informatica Cloud only uses the default date format when you:

- **Convert a date to a string by connecting a date/time field to a string field.** The service converts the date to a string in the default date format, MM/DD/YYYY HH24:MI:SS.

- **Convert a string to a date by connecting a string field to a date/time field.** The service expects the string values to be in the default date format, MM/DD/YYYY HH24:MI:SS. If an input value does not match this format, or it is an invalid date, Informatica Cloud skips the row. If the string is in the default date format, Informatica Cloud converts the string to a date value.

- **Use TO_CHAR(date, [format_string]) to convert dates to strings.** If you omit the format string, Informatica Cloud returns the string in the default date format, MM/DD/YYYY HH24:MI:SS. If you specify a format string, Informatica Cloud returns a string in the specified format.

- **Use TO_DATE(date, [format_string]) to convert strings to dates.** If you omit the format string, Informatica Cloud expects the string in the default date format, MM/DD/YYYY HH24:MI:SS. If you specify a format string, Informatica Cloud expects a string in the specified format.

The default date format of MM/DD/YYYY HH24:MI:SS consists of:

- Month (January = 01, September = 09)
- Day (of the month)
- Year (expressed in four digits, such as 1998)
- Hour (in 24-hour format, for example, 12:00:00AM = 0, 1:00:00AM = 1, 12:00:00PM = 12, 11:00:00PM = 23)
- Minutes
- Seconds

**Date Format Strings**

You can evaluate input dates using a combination of format strings and date functions. Date format strings are not internationalized and must be entered in predefined formats as listed in the following table.
The following table summarizes the format strings to specify a part of a date:

<table>
<thead>
<tr>
<th>Format String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D, DD, DDD, DAY, DY, J</td>
<td>Days (01-31). Use any of these format strings to specify the entire day portion of a date. For example, if you pass 12-APR-1997 to a date function, use any of these format strings specify 12.</td>
</tr>
<tr>
<td>HH, HH12, HH24</td>
<td>Hour of day (0-23), where 0 is 12 AM (midnight). Use any of these formats to specify the entire hour portion of a date. For example, if you pass the date 12-APR-1997 2:01:32 PM, use HH, HH12, or HH24 to specify the hour portion of the date.</td>
</tr>
<tr>
<td>MI</td>
<td>Minutes (0-59).</td>
</tr>
<tr>
<td>MM, MON, MONTH</td>
<td>Month (01-12). Use any of these format strings to specify the entire month portion of a date. For example, if you pass 12-APR-1997 to a date function, use MM, MON, or MONTH to specify APR.</td>
</tr>
<tr>
<td>SS, SSSS</td>
<td>Seconds (0-59).</td>
</tr>
<tr>
<td>Y, YY, YYYY, RR</td>
<td>Year portion of date (1753 to 9999). Use any of these format strings to specify the entire year portion of a date. For example, if you pass 12-APR-1997 to a date function, use Y, YY, YYYY, or YYYY to specify 1997.</td>
</tr>
</tbody>
</table>

**Note:** The format string is not case sensitive. It must always be enclosed within single quotation marks.

The following table uses date functions with date format strings to evaluate input dates:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD_TO_DATE</td>
<td>Specify the part of the date you want to change.</td>
</tr>
<tr>
<td>DATE_DIFF</td>
<td>Specify the part of the date to use to calculate the difference between two dates.</td>
</tr>
<tr>
<td>GET_DATE_PART</td>
<td>Specify the part of the date you want to return. This function returns an integer value based on the default date format.</td>
</tr>
<tr>
<td>ROUND</td>
<td>Specify the part of the date you want to round.</td>
</tr>
<tr>
<td>SET_DATE_PART</td>
<td>Specify the part of the date you want to change.</td>
</tr>
<tr>
<td>SYSTIMESTAMP</td>
<td>The timestamp precision.</td>
</tr>
<tr>
<td>TRUNC</td>
<td>Specify the part of the date you want to truncate.</td>
</tr>
</tbody>
</table>

**TO_CHAR Format Strings**

The TO_CHAR function converts a Date/Time datatype to a string with the format you specify. You can convert the entire date or a part of the date to a string. You might use TO_CHAR to convert dates to string, changing the format for reporting purposes.

TO_CHAR is generally used when the target is a flat file or a database that does not support a Date/Time datatype.
The following table summarizes the format strings for dates in the function TO_CHAR:

### Table 25. TO_CHAR Format Strings

<table>
<thead>
<tr>
<th>Format String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM, A.M., PM, P.M.</td>
<td>Meridian indicator. Use any of these format strings to specify AM and PM hours. AM and PM return the same values as A.M. and P.M.</td>
</tr>
<tr>
<td>D</td>
<td>Day of week (1-7), where Sunday equals 1.</td>
</tr>
<tr>
<td>DD</td>
<td>Day of month (01-31).</td>
</tr>
<tr>
<td>DDD</td>
<td>Day of year (001-366, including leap years).</td>
</tr>
<tr>
<td>DAY</td>
<td>Name of day, including up to nine characters (for example, Wednesday).</td>
</tr>
<tr>
<td>DY</td>
<td>Abbreviated three-character name for a day (for example, Wed).</td>
</tr>
<tr>
<td>HH, HH12</td>
<td>Hour of day (01-12).</td>
</tr>
<tr>
<td>HH24</td>
<td>Hour of day (00-23), where 00 is 12AM (midnight).</td>
</tr>
<tr>
<td>J</td>
<td>Modified Julian Day. Converts the calendar date to a string equivalent to its Modified Julian Day value, calculated from Jan 1, 4713 00:00:00 BC. It ignores the time component of the date. For example, the expression TO_CHAR(SHIP_DATE, 'J') converts Dec 31 1999 23:59:59 to the string 2451544.</td>
</tr>
<tr>
<td>MI</td>
<td>Minutes (00-59).</td>
</tr>
<tr>
<td>MM</td>
<td>Month (01-12).</td>
</tr>
<tr>
<td>MONTH</td>
<td>Name of month, including up to nine characters (for example, January).</td>
</tr>
<tr>
<td>MON</td>
<td>Abbreviated three-character name for a month (for example, Jan).</td>
</tr>
<tr>
<td>Q</td>
<td>Quarter of year (1-4), where January to March equals 1.</td>
</tr>
<tr>
<td>RR</td>
<td>Last two digits of a year. The function removes the leading digits. For example, if you use ‘RR’ and pass the year 1997, TO_CHAR returns 97. When used with TO_CHAR, ‘RR’ produces the same results as, and is interchangeable with, ‘YY.’ However, when used with TO_DATE, ‘RR’ calculates the closest appropriate century and supplies the first two digits of the year.</td>
</tr>
<tr>
<td>SS</td>
<td>Seconds (00-59).</td>
</tr>
<tr>
<td>SSSSS</td>
<td>Seconds since midnight (00000 - 86399). When you use SSSSS in a TO_CHAR expression, Informatica Cloud only evaluates the time portion of a date. For example, the expression TO_CHAR(SHIP_DATE, 'MM/DD/YYYY SSSSS') converts 12/31/1999 01:02:03 to 12/31/1999 03783.</td>
</tr>
<tr>
<td>Y</td>
<td>Last digit of a year. The function removes the leading digits. For example, if you use ‘Y’ and pass the year 1997, TO_CHAR returns 7.</td>
</tr>
<tr>
<td>YY</td>
<td>Last two digits of a year. The function removes the leading digits. For example, if you use “YY” and pass the year 1997, TO_CHAR returns 97.</td>
</tr>
</tbody>
</table>
### Format String

<table>
<thead>
<tr>
<th>Format String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YYY</td>
<td>Last three digits of a year. The function removes the leading digits. For example, if you use &quot;YYY&quot; and pass the year 1997, TO_CHAR returns 997.</td>
</tr>
<tr>
<td>YYYY</td>
<td>Entire year portion of date. For example, if you use &quot;YYYY&quot; and pass the year 1997, TO_CHAR returns 1997.</td>
</tr>
<tr>
<td>W</td>
<td>Week of month (1-5), where week 1 starts on the first day of the month and ends on the seventh, week 2 starts on the eighth day and ends on the fourteenth day. For example, Feb 1 designates the first week of February.</td>
</tr>
<tr>
<td>WW</td>
<td>Week of year (01-53), where week 01 starts on Jan 1 and ends on Jan 7, week 2 starts on Jan 8 and ends on Jan 14, and so on.</td>
</tr>
<tr>
<td>- / . ; :</td>
<td>Punctuation that displays in the output. You might use these symbols to separate date parts. For example, you might create the following expression to separate date parts with a period: TO_CHAR( DATES, 'MM.DD.YYYY' ).</td>
</tr>
<tr>
<td>&quot;text&quot;</td>
<td>Text that displays in the output. For example, if you have the expression: TO_CHAR( DATES, 'MM/DD/YYYY &quot;Sales Were Up&quot;' ) and pass the date Apr 1 1997, the function returns the string '04/01/1997 Sales Were Up'. You can enter multibyte characters that are valid in the repository code page.</td>
</tr>
<tr>
<td>**</td>
<td>Use double quotation marks to separate ambiguous format strings, for example D'D&quot;DDD. The empty quotation marks do not appear in the output.</td>
</tr>
</tbody>
</table>

**Note:** The format string is not case sensitive. It must always be enclosed within single quotation marks.

### Examples

The following examples illustrate the J, SSSSS, RR, and YY format strings. See the individual functions for more examples.

**Note:** The service ignores the time portion of the date in a TO_CHAR expression.

#### J Format String

Use the J format string in a TO_CHAR expression to convert date values to MJD values expressed as strings. For example:

```sql
TO_CHAR(SHIP_DATE, "J")
```

<table>
<thead>
<tr>
<th>SHIP_DATE</th>
<th>RETURN_VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dec 31 1999 23:59:59</td>
<td>2451544</td>
</tr>
<tr>
<td>Jan 1 1900 01:02:03</td>
<td>2415021</td>
</tr>
</tbody>
</table>

#### SSSSS Format String

You can also use the format string SSSSS in a TO_CHAR expression. For example, the following expression converts the dates in the SHIP_DATE port to strings representing the total seconds since midnight:

```sql
TO_CHAR( SHIP_DATE, 'SSSSS')
```

<table>
<thead>
<tr>
<th>SHIP_DATE</th>
<th>RETURN_VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/31/1999 01:02:03</td>
<td>3783</td>
</tr>
<tr>
<td>09/15/1996 23:59:59</td>
<td>86399</td>
</tr>
</tbody>
</table>

308 Appendix E: Dates
RR Format String

The following expression converts dates to strings in the format MM/DD/YY:

```
TO_CHAR( SHIP_DATE, 'MM/DD/RR')
```

<table>
<thead>
<tr>
<th>SHIP_DATE</th>
<th>RETURN_VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/31/1999 01:02:03</td>
<td>12/31/99</td>
</tr>
<tr>
<td>09/15/1996 23:59:59</td>
<td>09/15/96</td>
</tr>
<tr>
<td>05/17/2003 12:13:14</td>
<td>05/17/03</td>
</tr>
</tbody>
</table>

YY Format String

In TO_CHAR expressions, the YY format string produces the same results as the RR format string. The following expression converts dates to strings in the format MM/DD/YY:

```
TO_CHAR( SHIP_DATE, 'MM/DD/YY')
```

<table>
<thead>
<tr>
<th>SHIP_DATE</th>
<th>RETURN_VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/31/1999 01:02:03</td>
<td>12/31/99</td>
</tr>
<tr>
<td>09/15/1996 23:59:59</td>
<td>09/15/96</td>
</tr>
<tr>
<td>05/17/2003 12:13:14</td>
<td>05/17/03</td>
</tr>
</tbody>
</table>

TO_DATE and IS_DATE Format Strings

The TO_DATE function converts a string with the format you specify to a datetime value. TO_DATE is generally used to convert strings from flat files to datetime values. TO_DATE format strings are not internationalized and must be entered in predefined formats as listed in the table below.

**Note:** TO_DATE and IS_DATE use the same set of format strings.

When you create a TO_DATE expression, use a format string for each part of the date in the source string. The source string format and the format string must match, including any date separators. If any parts do not match, Informatica Cloud does not convert the string and skips the row. If you omit the format string, the source string must be in the default date format MM/DD/YYYY HH24:MI:SS.

IS_DATE tells you if a value is a valid date. A valid date is any string representing a valid date in the default date format of MM/DD/YYYY HH24:MI:SS. If the strings you want to test are not in the default date format, use the format strings listed in the TO_DATE and IS_DATE format Strings table to specify the date format. If a string does not match the specified format string or is not a valid date, the function returns FALSE (0). If the string matches the format string and is a valid date, the function returns TRUE (1). IS_DATE format strings are not internationalized and must be entered in predefined formats as listed in the following table.

The following table summarizes the format strings for the functions TO_DATE and IS_DATE:

### Table 26. TO_DATE and IS_DATE Format Strings

<table>
<thead>
<tr>
<th>Format String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM, a.m., PM, p.m.</td>
<td>Meridian indicator. Use any of these format strings to specify AM and PM hours. AM and PM return the same values as do a.m. and p.m.</td>
</tr>
<tr>
<td>DD</td>
<td>Day of month (1-31).</td>
</tr>
<tr>
<td>DDD</td>
<td>Day of year (001-366, including leap years).</td>
</tr>
<tr>
<td>Format String</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>DAY</td>
<td>Name of day, including up to nine characters (for example, Wednesday). The DAY format string is not case sensitive.</td>
</tr>
<tr>
<td>DY</td>
<td>Abbreviated three-character name for a day (for example, Wed). The DY format string is not case sensitive.</td>
</tr>
<tr>
<td>HH, HH12</td>
<td>Hour of day (1-12).</td>
</tr>
<tr>
<td>HH24</td>
<td>Hour of day (0-23), where 0 is 12AM (midnight).</td>
</tr>
<tr>
<td>MI</td>
<td>Minutes (0-59).</td>
</tr>
</tbody>
</table>
| RR            | Four-digit year (for example, 1998, 2034). Use when source strings include two-digit years. Use with TO_DATE to convert two-digit years to four-digit years.  
- Current Year Between 50 and 99. If the current year is between 50 and 99 (such as 1998) and the year value of the source string is between 0 and 49, Informatica Cloud returns the next century plus the two-digit year from the source string. If the year value of the source string is between 50 and 99, Informatica Cloud returns the current century plus the specified two-digit year.  
- Current Year Between 0 and 49. If the current year is between 0 and 49 (such as 2003) and the source string year is between 0 and 49, Informatica Cloud returns the current century plus the two-digit year from the source string. If the source string year is between 50 and 99, Informatica Cloud returns the previous century plus the two-digit year from the source string. |
| SS            | Seconds (0-59). |
| SSSSS         | Seconds since midnight. When you use SSSSS in a TO_DATE expression, Informatica Cloud only evaluates the time portion of a date.  
For example, the expression TO_DATE( DATE_STR, 'MM/DD/YYYY SSSSS') converts 12/31/1999 3783 to 12/31/1999 01:02:03. |
| J             | Modified Julian Day. Convert strings in MJD format to date values. It ignores the time component of the source string, assigning all dates the time of 00:00:00.  
For example, the expression TO_DATE('2451544', 'J') converts 2451544 to Dec 31 1999 00:00:00. |
| MM            | Month (1-12). |
| MONTH         | Name of month, including up to nine characters (for example, August). Case does not matter. |
| MON           | Abbreviated three-character name for a month (for example, Aug). Case does not matter. |
| NS            | Nanoseconds. TO_DATE and IS_DATE can support sub-seconds by using the format token 'NS'. The unit is nanosecond. If the sub-second portion is in milliseconds, you can still use it by appending three zeroes as shown in the following examples:  
TO_DATE('2005-05-02 09:23:34.123' || '000', 'YYYY-MM-DD HH24:MI:SS.NS')  
<p>| Y             | The current year on the machine running the Secure Agent with the last digit of the year replaced with the string value. |
| YY            | The current year on the machine running the Secure Agent with the last two digits of the year replaced with the string value. |</p>
<table>
<thead>
<tr>
<th>Format String</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>YYY</td>
<td>The current year on the machine running the Secure Agent with the last three digits of the year replaced with the string value.</td>
</tr>
<tr>
<td>YYYY</td>
<td>Four digits of a year. Do not use this format string if you are passing two-digit years. Use the RR or YY format string instead.</td>
</tr>
</tbody>
</table>

**Requirements**

Informatica Cloud expects the format of the TO_DATE string to meet the following conditions:

- The format of the TO_DATE string must match the format string including any date separators. If it does not, Informatica Cloud might return inaccurate values or skip the row. For example, if you pass the string ‘20200512’, representing May 12, 2020, to TO_DATE, you must include the format string YYYYYMDD. If you do not include a format string, Informatica Cloud expects the string in the default date format MM/DD/YYYY HH24:MI:SS. Likewise, if you pass a string that does not match the format string, Informatica Cloud returns an error and skips the row. For example, if you pass the string 2020120 to TO_DATE and include the format string YYYYYMDD, Informatica Cloud returns an error and skips the row because the string does not match the format string.

- The format string must always be enclosed within single quotation marks.

**Tip:** By default, Informatica Cloud uses the format string MM/DD/YYYY HH24:MI:SS. The format string is not case sensitive.

**Example**

The following examples illustrate the J, RR, and SSSSS format strings. See the individual functions for more examples.

**J Format String**

The following expression converts strings in the SHIP_DATE_MJD_STRING field to date values in the default date format:

```
TO_DATE (SHIP_DATE_MJD_STR, 'J')
```

<table>
<thead>
<tr>
<th>SHIP_DATE_MJD_STR</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2451544</td>
<td>Dec 31 1999 00:00:00</td>
</tr>
<tr>
<td>245021</td>
<td>Jan 1 1900 00:00:00</td>
</tr>
</tbody>
</table>

Because the J format string does not include the time portion of a date, the return values have the time set to 00:00:00.

**RR Format String**

The following expression converts a string to a four-digit year format. The current year is 1998:

```
TO_DATE( DATE_STR, "MM/DD/RR")
```

<table>
<thead>
<tr>
<th>DATE_STR</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/07/98</td>
<td>04/01/1998 00:00:00</td>
</tr>
<tr>
<td>08/17/05</td>
<td>08/17/2005 00:00:00</td>
</tr>
</tbody>
</table>
YY Format String

The following expression converts a string to a four-digit year format. The current year is 1998:

\[
\text{TO\_DATE( DATE\_STR, 'MM/DD/YY')}
\]

<table>
<thead>
<tr>
<th>DATE_STR</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/07/98</td>
<td>04/01/1998 00:00:00</td>
</tr>
<tr>
<td>08/17/05</td>
<td>08/17/1905 00:00:00</td>
</tr>
</tbody>
</table>

**Note:** For the second row, RR returns the year 2005, but YY returns the year 1905.

SSSSS Format String

The following expression converts strings that include the seconds since midnight to date values:

\[
\text{TO\_DATE( DATE\_STR, 'MM/DD/YYYY SSSSS')}
\]

<table>
<thead>
<tr>
<th>DATE_STR</th>
<th>RETURN VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/31/1999 3783</td>
<td>12/31/1999 01:02:03</td>
</tr>
</tbody>
</table>

Understanding Date Arithmetic

The transformation language provides built-in date functions so you can perform arithmetic on datetime values as follows:

- **ADD\_TO\_DATE.** Add or subtract a specific portion of a date.
- **DATE\_DIFF.** Subtract two dates.
- **SET\_DATE\_PART.** Change one part of a date.

You cannot use numeric arithmetic operators (such as + or -) to add or subtract dates.

Informatica Cloud recognizes leap years and accepts dates between Jan. 1, 1753 00:00:00 AD and Dec. 31, 9999 23:59:59 AD.

**Note:** Informatica Cloud uses the Date/Time datatype to specify date values. You can only use the date functions on datetime values.
A

activity log
A log that contains log entries for all tasks that have completed. If a job runs successfully, the activity log indicates the number of processed rows. If the job fails or completes with errors, the activity log displays related error messages.

activity monitor
A component of Informatica Cloud that provides details about tasks that are currently running.

administrator
See organization administrator on page 315 or Informatica Cloud Administrator on page 315.

agent
See Informatica Cloud Secure Agent on page 315.

audit log
A log that tracks user operations such as login, create, update, and delete operations.

C

custom source
An Informatica Cloud object that you can create from one or more source database tables.

custom view
A custom view is a view that displays a subset of the specified object type. You can configure custom views for connections, tasks, and task flows.

D

dashboard
A page where you view the scorecard for each data assessment task that has completed and that is configured to show results in the dashboard.
Data Assessment service
An Informatica Cloud service that measures the quality of data in Salesforce.

Data Loader service
An Informatica Cloud service that reads data from and writes data to Salesforce, databases, and flat files.

Data Replication service
An Informatica Cloud service that replicates data from Salesforce objects to a target. You might replicate data to archive data, perform offline reporting, or consolidate and manage data.

Data Synchronization service
An Informatica Cloud service that reads data from sources and writes data to targets. The Data Synchronization service contains all Data Loader service functionality. It also contains additional functionality, such as using multiple source objects, custom source objects, multiple tasks in a schedule, and additional source and target combinations.

email notification
Email messages that Informatica Cloud sends to inform users about events. You can configure Informatica Cloud to send notification when tasks complete and when passwords are reset.

object-level permissions
Object-level permissions allow you to configure user group access for individual objects in the organization.

full load
A type of load where the Data Replication service replicates the data for all rows of the Salesforce objects in the data replication task. Each time the data replication task runs, the Data Replication service truncates the target database tables and performs a full data refresh from Salesforce.

hosting facility
See Informatica Cloud hosting facility on page 315.

incremental load
A type of load where the Data Replication service replicates the data for new rows and the rows that have changed since the last replication. It does not detect or delete existing rows in the target database that are deleted in the Salesforce source.
Informatica Cloud Administrator

The Informatica Cloud Administrator is a browser-based application that allows you to manage an organization hierarchy. Use the Informatica Cloud Administrator to create sub-organizations and to perform administrative tasks within the organization hierarchy.

Informatica Cloud application

A browser-based application that runs at the Informatica Cloud hosting facility. It allows you to configure connections, create users, and create, run, schedule, and monitor tasks.

Informatica Cloud REST API

A Representational State Transfer (REST) application programming interface (API) for Informatica Cloud.

Informatica Cloud Services

Informatica Cloud services that perform tasks. Informatica Cloud Services include the Data Loader service, Data Replication service, Data Assessment service, Data Synchronization service, and PowerCenter service.

Informatica Cloud hosting facility

A facility where Informatica Cloud runs. It connects to your organization and Salesforce to process service requests.

Informatica Cloud Secure Agent

A lightweight program that enables secure communication across the firewall between your organization and Informatica Cloud. Also known as the Secure Agent.

Informatica Cloud Secure Agent Manager

A program used to manage the Informatica Cloud Secure Agent on Windows.

J

job

A job is an instance of a task.

M

maximum daily jobs

An organization property that indicates maximum number of jobs an organization can run every day. This includes all jobs that start, regardless of whether they complete successfully.

O

organization administrator

An Informatica Cloud user configured with the Admin role and full access to the organization. The organization administrator is responsible for managing the organization and performing administrative tasks, such as configuring Secure Agents, configuring the organization profile, and managing organization security. If the organization has multiple users with the Admin role, the organization administrator is the lead administrator.
plug-in
Transformation logic that you can create and use to transform source data before it is loaded into the target. Previously known as a custom function.

PowerCenter service
An Informatica Cloud service that runs a PowerCenter workflow as an Informatica Cloud PowerCenter task.

quality metric
A metric that performs a specific type of data quality analysis. For example, the Address Validation quality metric determines the percentage of address-type fields that have valid address data.

role
A role is a set of privileges that allows a user to access information and perform tasks in the organization. Each user must be assigned at least one role.

scorecard
A summary of the results of quality metrics included in the data assessment task. When you run the data assessment task, the Data Assessment service runs the quality check for each data quality metric included in the plan and publishes the results in a scorecard.

sub-organization
An Informatica Cloud organization that is part of an organization hierarchy.

task
A process that you configure to analyze, extract, transform, and load data. Informatica Cloud allows you to configure and run different types of tasks. For example, you can run a data assessment task that measures field completeness, field conformance, record duplication, and address validity for each object in a Salesforce account.

task flow
An Informatica Cloud object that allows you to group multiple tasks. You can run the task flow immediately or on a schedule. The tasks in a task flow run serially, in the specified order.

user
A user is an Informatica Cloud user account that allows secure access to an organization. A user can perform tasks in the organization based on the role, user group and permissions assigned to the user.
user group

A user group is a set of permissions that allows a user to perform tasks on objects in the organization.
Informatica Cloud REST API Overview

The Informatica Cloud REST API allows you to access information from your Informatica Cloud organization, create and update users, start tasks and task flows, and monitor job activity.

To use the Informatica Cloud REST API, you need a valid Informatica Cloud login and knowledge of general REST API guidelines.

To perform a task using the Informatica Cloud REST API, use the appropriate resource and method, along with the applicable parameters. Informatica Cloud returns the requested information, performs the requested task, or returns a response code and related messages.

The Informatica Cloud REST API provides the following resources:
- **activitylog**: Returns job details from the Informatica Cloud activity log.
- **activitymonitor**: Returns job details from the Informatica Cloud activity monitor.
- **dnbbatch**: Allows you to view or update the definition of a D&B360 batch. (Available for Informatica Cloud D&B360 partners only.)
REST API Response Formats

The Informatica Cloud REST API supports responses in JSON and XML. By default, responses are in XML.

The following code is a sample response in XML:

```xml
<?xml version="1.0" encoding="utf-8" ?>
<LoginResponse>
  <errorCode>NO_ERROR</errorCode>
  <errorMsg />
  <sessionId>bZTxMxokc877HFBSIzA+bu1Gs/HcQB3pgKJ/pRWfwCJEXIGBr0IYhaenAzFAm6xH</sessionId>
</LoginResponse>
```

The following code is the same response in JSON:

```json
{"errorCode":"NO_ERROR", "errorMsg":null, "sessionId":"mZKIbi55bNO5m99989X0y8THdSdmwp00P54aq2c9uCpLot12jim1JMLblYwYa0L"}
```

XML Schema Definition

You can access the Informatica Cloud API XML schema definition (XSD) at the following URL:

https://app.informaticaondemand.com/saas/api/ics_api_1.xsd

To use the XSD URL, log into Informatica Cloud, then enter the XSD URL.

API Test Page

Informatica Cloud provides a test page where you can test the resources, methods, and parameters you want to use. You can access the test page at the following URL:

https://app.informaticaondemand.com/saas/app/admin/apiTest.do

To use the test page URL, log into Informatica Cloud, then enter the test page URL.

Documentation Conventions

Informatica Cloud REST API documentation uses the following conventions:

- Methods are in capital letters, such as GET.
- Syntax descriptions use the following conventions:
  - The resource name is separated from related parameters by a question mark (?).
  - Parameters are separated by ampersands (&).
  - Parameter values are enclosed in angle brackets (< >).
  - When listing a choice of parameter values, options are separated by a pipe (|).
- Optional parameters are in italics.

In the following example, the resource is runjob, the parameters are icSessionId, jobName, jobType, and responseType. The responseType parameter is optional, and possible values for the parameter are xml or json:

```
runjob?icSessionId=<IC session ID>&jobName=<task or taskflow name>&jobType=<task type>&responseType=<xml|json>
```

REST API Guidelines

Use the following guidelines when working with Informatica Cloud REST API:

- When constructing a request, configure the resource and parameters in the format you want to use for submission.
- If you do not specify a response type, Informatica Cloud returns responses in XML. To return results in JSON, set the responseType parameter to json.
- All resources and parameters are case sensitive.
- Use the following URL as the base for all resource calls:
  
  https://app.informaticaondemand.com/saas/api/1/

GET Example

The following code displays how you might use the activitymonitor resource to view information from the activity monitor:

```java
public void listActivityLog(String icSessionId, boolean details) {
    try {
        //Open connection for request
        URL url = new URL("https://app.informaticaondemand.com/saas/api/1/activitymonitor");
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");
        conn.setDoOutput(true);
        OutputStreamWriter writer = new OutputStreamWriter(conn.getOutputStream());
        //write parameters
        String responseType = "XML";
        String sessionId, taskId, jobName,
        taskType, icSessionId, details = "true";
        String[] urlParameters = {"responseType=", "icSessionId=", "details="};
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < urlParameters.length; i++) {
            sb.append(urlParameters[i]);
        }
        writer.write(sb.toString());
        writer.flush();

        //Get the response
        String response = new String(conn.getInputStream().readAllBytes());
        System.out.println(response);
    } catch (MalformedURLException ex) {
        ex.printStackTrace();
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}
```
**POST Example**

The following code displays how you might use the login resource to log in and generate an Informatica Cloud session ID:

```java
public void login(String username, String password) {
    try {
        // Open connection for request
        URL url = new URL("https://app.informaticaonondemand.com/saas/api/1/login");
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("POST");
        conn.setDoOutput(true);
        OutputStreamWriter writer = new OutputStreamWriter(conn.getOutputStream());

        // write parameters
        String responseType = "XML";
        StringBuilder sb = new StringBuilder();
        sb.append("responseType=").append(responseType);
        sb.append("&username=").append(username);
        sb.append("&password=").append(password);
        writer.write(sb.toString());
        writer.flush();

        // Get the response
        StringBuffer answer = new StringBuffer();
        BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
        String line;
        while ((line = reader.readLine()) != null) {
            answer.append(line);
        }
        writer.close();
        reader.close();

        // Output the response
        System.out.println(answer.toString());
    } catch (MalformedURLException ex) {
        ex.printStackTrace();
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}
```

**Response Codes**

Informatica Cloud REST API provides the following response codes:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO_ERROR</td>
<td>The request completed without errors.</td>
</tr>
<tr>
<td>ACTIVITY_LOG_ERROR</td>
<td>An error occurred while accessing the activity log.</td>
</tr>
<tr>
<td>ACTIVITY_MONITOR_ERROR</td>
<td>An error occurred while accessing the activity monitor.</td>
</tr>
<tr>
<td>DNB_BATCH_ERROR</td>
<td>An error occurred while processing a D&amp;B360 batch.</td>
</tr>
<tr>
<td>DNB_BATCH_SFLOGIN_ERROR</td>
<td>You used an invalid Salesforce login with the dnbbatch resource.</td>
</tr>
<tr>
<td>INVALID_SESSION_ID</td>
<td>You used an invalid Informatica Cloud or Salesforce session ID.</td>
</tr>
<tr>
<td>INSUFFICIENT_PRIVILEGE</td>
<td>You do not have the appropriate privilege to perform the request.</td>
</tr>
<tr>
<td>Return Code</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>INSUFFICIENT_PERMISSION</td>
<td>You do not have the appropriate permissions to perform the request.</td>
</tr>
<tr>
<td>LOGIN_AUTHENTICATION_FAILED</td>
<td>Informatica Cloud was unable to authenticate the login. Verify that you are using the correct user name and password combination.</td>
</tr>
<tr>
<td>LOGIN_INVALID_USERNAME</td>
<td>You entered an invalid user name.</td>
</tr>
<tr>
<td>LOGIN_INVALID_SALESFORCE_SESSION</td>
<td>You entered an invalid Salesforce session ID.</td>
</tr>
<tr>
<td>REGISTER_DUPLICATE_USERNAME</td>
<td>The user name you want to create is already in use.</td>
</tr>
<tr>
<td>REGISTER_ERROR</td>
<td>An error occurred while creating an Informatica Cloud organization.</td>
</tr>
<tr>
<td>REGISTER_INVALID_OFFER_CODE</td>
<td>You entered an invalid partner offer code.</td>
</tr>
<tr>
<td>RUN_JOB_ERROR</td>
<td>An error occurred while starting a task or task flow.</td>
</tr>
<tr>
<td>USER_ERROR</td>
<td>An error occurred while using the user resource.</td>
</tr>
<tr>
<td>UNSUPPORTED_METHOD</td>
<td>You used a method that is not supported in this version of the Informatica Cloud REST API.</td>
</tr>
</tbody>
</table>

### Resource Quick Reference

The following table contains the syntax and a brief description of the Informatica Cloud REST API resources:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Method</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>activitylog</td>
<td>GET</td>
<td>activitylog?icSessionId=&lt;IC session ID&gt; &amp;rowLimit=&lt;number of rows&gt; &amp;responseType=&lt;xml</td>
<td>json&gt;</td>
</tr>
<tr>
<td>activitymonitor</td>
<td>GET</td>
<td>activitymonitor?icSessionId=&lt;IC session ID&gt; &amp;details=&lt;true</td>
<td>false&gt; &amp;responseType=&lt;xml</td>
</tr>
<tr>
<td>dnbbatch</td>
<td>GET</td>
<td>dnbbatch?icSessionId=&lt;IC session ID&gt; &amp;responseType=&lt;xml</td>
<td>json&gt;</td>
</tr>
<tr>
<td>POST</td>
<td></td>
<td>dnbbatch?icSessionId=&lt;IC session ID&gt; &amp;sfUsername=&lt;Salesforce user name&gt; &amp;sfPassword=&lt;Salesforce password&gt; &amp;sfSecurityToken=&lt;Salesforce security token&gt; &amp;rowLimit=&lt;row limit&gt; &amp;enableNewBatch=&lt;new batch enabled&gt; &amp;newBatchStartTime=&lt;new batch start time&gt; &amp;newBatchRepeatFrequency=&lt;new batch repeat frequency&gt; &amp;newBatchRepeatInterval=&lt;</td>
<td>Updates the details of a D&amp;B360 batch. For more information, see “dnbbatch” on page 327.</td>
</tr>
<tr>
<td>Resource</td>
<td>Method</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>&amp;&lt;new batch repeat interval&gt;&amp;enableRefreshBatch=&lt;refresh batch enabled&gt; &amp;refreshBatchStartTime=&lt;refresh batch start time&gt; &amp;refreshBatchRepeatFrequency=&lt;refresh batch repeat frequency&gt; &amp;refreshBatchRepeatInterval=&lt;refresh batch repeat interval&gt; &amp;responseType=&lt;xml</td>
<td>json&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>login</td>
<td>POST</td>
<td>login?username=&lt;IC username&gt; &amp;password=&lt;IC password&gt; &amp;responseType=&lt;xml</td>
<td>json&gt; or login?sfSessionId=&lt;Salesforce Session ID&gt; &amp;sfServerUrl=&lt;Salesforce URL&gt; &amp;responseType=&lt;xml</td>
</tr>
<tr>
<td>register</td>
<td>POST</td>
<td>register?offerCode=&lt;offer code&gt; &amp;sfSessionId=&lt;Salesforce session ID&gt; &amp;sfServerUrl=&lt;Salesforce URL&gt; &amp;responseType=&lt;xml</td>
<td>json&gt; or register?offerCode=&lt;offer code&gt; &amp;orgName&lt;organization name&gt; &amp;firstName=&lt;first name&gt; &amp;lastName=&lt;last name&gt; &amp;title=&lt;title&gt; &amp;email=&lt;email address&gt; &amp;phone=&lt;phone number&gt;&amp;address=&lt;address&gt; &amp;city=&lt;city&gt; &amp;state=&lt;state&gt; &amp;zipCode=&lt;zip code&gt; &amp;country=&lt;country&gt; &amp;employees=&lt;employees&gt; &amp;responseType=&lt;xml</td>
</tr>
<tr>
<td>runjob</td>
<td>POST</td>
<td>runjob?icSessionId=&lt;IC session ID&gt;= &amp;jobName=&lt;task or taskflow name&gt; &amp;jobType=&lt;task type&gt; &amp;responseType=&lt;xml</td>
<td>json&gt; or runjob?username=&lt;IC user name&gt; &amp;password=&lt;IC password&gt; &amp;jobName=&lt;task or taskflow name&gt; &amp;jobType=&lt;task type&gt; &amp;responseType=&lt;xml</td>
</tr>
<tr>
<td>salesforceversion</td>
<td>GET</td>
<td>salesforceversion?responseType=&lt;xml</td>
<td>json&gt;</td>
</tr>
<tr>
<td>servertime</td>
<td>GET</td>
<td>servertime?responseType=&lt;xml</td>
<td>json&gt;</td>
</tr>
<tr>
<td>user</td>
<td>GET</td>
<td>user?icSessionId=&lt;IC session ID&gt; &amp;username=&lt;IC user name&gt; &amp;responseType=&lt;xml</td>
<td>json&gt; or user?icSessionId=&lt;IC session ID&gt; &amp;username=&lt;IC user name&gt; &amp;sfUsername=&lt;Salesforce user name&gt; &amp;firstName=&lt;first name&gt; &amp;lastName=&lt;last name&gt; &amp;title=&lt;title&gt; &amp;phone=&lt;phone&gt;</td>
</tr>
<tr>
<td></td>
<td>POST</td>
<td>user?icSessionId=&lt;IC session ID&gt; &amp;username=&lt;IC user name&gt; &amp;sfUsername=&lt;Salesforce user name&gt; &amp;firstName=&lt;first name&gt; &amp;lastName=&lt;last name&gt; &amp;title=&lt;title&gt; &amp;phone=&lt;phone&gt;</td>
<td>Updates the details of an Informatica Cloud user account. For more information, see “user” on page 335.</td>
</tr>
<tr>
<td>Resource</td>
<td>Method</td>
<td>Syntax</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&amp;description=&lt;description&gt;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&amp;responseType=&lt;xml</td>
<td>json&gt;</td>
</tr>
</tbody>
</table>

**activitylog**

Use this resource to return information from the activity log.

**GET Syntax**

```
activitylog?icSessionId=<IC session ID>&rowLimit=<number of rows>&responseType=<xml|json>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>icSessionId</td>
<td>Required</td>
<td>The Informatica Cloud REST API session ID.</td>
</tr>
<tr>
<td>rowLimit</td>
<td>Optional</td>
<td>The maximum number of rows to return.</td>
</tr>
<tr>
<td>responseType</td>
<td>Optional</td>
<td>Format for the response from Informatica Cloud. Use one of the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- json. Returns information in JSON.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- xml. Returns information in XML.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>By default, Informatica Cloud responds in XML.</td>
</tr>
</tbody>
</table>

**Return**

Returns the following information from the activity log:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Task type. Returns one of the following codes:</td>
</tr>
<tr>
<td></td>
<td>- DLS. Data loader task.</td>
</tr>
<tr>
<td></td>
<td>- DSS. Data synchronization task.</td>
</tr>
<tr>
<td></td>
<td>- DRS. Data replication task.</td>
</tr>
<tr>
<td></td>
<td>- DQA. Data assessment task.</td>
</tr>
<tr>
<td></td>
<td>- PCS. PowerCenter task.</td>
</tr>
<tr>
<td></td>
<td>- Workflow. Task flow.</td>
</tr>
<tr>
<td></td>
<td>- DNB_Task. D&amp;B360 task.</td>
</tr>
<tr>
<td></td>
<td>- DNB_Workflow. D&amp;B360 workflow.</td>
</tr>
<tr>
<td>objectName</td>
<td>Task name.</td>
</tr>
<tr>
<td>runId</td>
<td>ID for the task run.</td>
</tr>
<tr>
<td>startTime</td>
<td>Start time for the task or task flow.</td>
</tr>
<tr>
<td>endTime</td>
<td>End time for the task or task flow.</td>
</tr>
<tr>
<td>success</td>
<td>Whether the task completed successfully. Returns one of the following codes:</td>
</tr>
<tr>
<td></td>
<td>- true. The task completed successfully or with errors.</td>
</tr>
<tr>
<td></td>
<td>- false. The task failed to complete.</td>
</tr>
<tr>
<td>Element Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>failedSourceRows</td>
<td>Number of rows that were not read from the source.</td>
</tr>
<tr>
<td>successSourceRows</td>
<td>Number of rows that were successfully read from the source.</td>
</tr>
<tr>
<td>failedTargetRows</td>
<td>Number of rows that were not written to the target.</td>
</tr>
<tr>
<td>successTargetRows</td>
<td>Number of rows that were successfully written to the target.</td>
</tr>
<tr>
<td>errorMsg</td>
<td>Error message associated with the job.</td>
</tr>
<tr>
<td>entries</td>
<td>Indicates the start of information for a child object. A child object might be a task within a task flow, or an object in a data replication task or D&amp;B360 workflow.</td>
</tr>
</tbody>
</table>

The following is a sample return in XML:

```xml
<?xml version="1.0" encoding="utf-8"?>
<ActivityLogResponse>
<errorCode>NO_ERROR</errorCode>
<errorMsg></errorMsg>
<entries>
    <ActivityLogEntry>
        <type>WORKFLOW</type>
        <objectName>task flow</objectName>
        <runId>1</runId>
        <startTime>2011-05-06T19:38:38.000Z</startTime>
        <endTime>2011-05-06T19:39:35.000Z</endTime>
        <success>1</success>
        <failedSourceRows>0</failedSourceRows>
        <successSourceRows>60</successSourceRows>
        <failedTargetRows>0</failedTargetRows>
        <successTargetRows>60</successTargetRows>
    </ActivityLogEntry>
    <ActivityLogEntry>
        <type>DRS</type>
        <objectName>DRS</objectName>
        <runId>9</runId>
        <startTime>2011-05-06T19:38:38.000Z</startTime>
        <endTime>2011-05-06T19:39:16.000Z</endTime>
        <success>1</success>
        <failedSourceRows>0</failedSourceRows>
        <successSourceRows>26</successSourceRows>
        <failedTargetRows>0</failedTargetRows>
        <successTargetRows>26</successTargetRows>
    </ActivityLogEntry>
    <ActivityLogEntry>
        <type>DRS</type>
        <objectName>Lead</objectName>
        <runId>0</runId>
        <startTime>2011-05-06T19:38:42.000Z</startTime>
        <endTime>2011-05-06T19:38:51.000Z</endTime>
        <success>1</success>
        <failedSourceRows>0</failedSourceRows>
        <successSourceRows>22</successSourceRows>
        <failedTargetRows>0</failedTargetRows>
        <successTargetRows>22</successTargetRows>
        <errorMsg>No errors encountered.</errorMsg>
    </ActivityLogEntry>
</entries>
</ActivityLogResponse>
```

**Example**

To return 20 rows from the activity log, you might use the following URL and parameters:

```plaintext
https://app.informaticaondemand.com/saas/api/1/activitylog
Parameters:
icSessionId=r2IWJdXxrfKgTEcKgGc9Duc6shnZk2ggJdgIRjCB7XX5vAsjLSZxcJ1NQK5MEECf
rowLimit=20
```

Note that because the response type was not defined, Informatica Cloud returns the response in XML.
activitymonitor

Use this resource to return information from the activity monitor.

GET Syntax

activitymonitor?icSessionId=<IC session ID>&details=<true|false>&responseType=<xml|json>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>icSessionId</td>
<td>Required</td>
<td>The Informatica Cloud REST API session ID.</td>
</tr>
<tr>
<td>details</td>
<td>Optional</td>
<td>Activity monitor detail to be returned from Informatica Cloud. Use one of the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- true. Returns activity monitor information for tasks and task flows. Also returns details for child objects, such as tasks within task flows, and the objects within data replication tasks and D&amp;B360 workflows.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- false. Returns activity monitor information for tasks and task flows. By default, activitymonitor returns information for tasks and task flows, without additional details about child objects.</td>
</tr>
<tr>
<td>responseType</td>
<td>Optional</td>
<td>Format for the response from Informatica Cloud. Use one of the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- json. Returns information in JSON.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- xml. Returns information in XML.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>By default, Informatica Cloud responds in XML.</td>
</tr>
</tbody>
</table>

Return

The following information about tasks and task flows from the activity monitor:

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Task type. Returns one of the following codes:</td>
</tr>
<tr>
<td></td>
<td>- DLS. Data loader task.</td>
</tr>
<tr>
<td></td>
<td>- DSS. Data synchronization task.</td>
</tr>
<tr>
<td></td>
<td>- DRS. Data replication task.</td>
</tr>
<tr>
<td></td>
<td>- DQA. Data assessment task.</td>
</tr>
<tr>
<td></td>
<td>- PCS. PowerCenter task.</td>
</tr>
<tr>
<td></td>
<td>- Workflow. Task flow.</td>
</tr>
<tr>
<td></td>
<td>- DNB_Task. D&amp;B360 task.</td>
</tr>
<tr>
<td></td>
<td>- DNB_Workflow. D&amp;B360 workflow.</td>
</tr>
<tr>
<td>taskName</td>
<td>Task name.</td>
</tr>
<tr>
<td>objectName</td>
<td>Source object used in the task, or the data replication or D&amp;B360 object being processed.</td>
</tr>
<tr>
<td>runId</td>
<td>ID for the task run.</td>
</tr>
<tr>
<td>startTime</td>
<td>Start time for the task or task flow.</td>
</tr>
<tr>
<td>endTime</td>
<td>End time for the task or task flow.</td>
</tr>
<tr>
<td>Element Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>executionState</td>
<td>State of the task. Returns one of the following codes:</td>
</tr>
<tr>
<td></td>
<td>- INITIALIZED</td>
</tr>
<tr>
<td></td>
<td>- RUNNING</td>
</tr>
<tr>
<td></td>
<td>- STOPPING</td>
</tr>
<tr>
<td></td>
<td>- COMPLETED</td>
</tr>
<tr>
<td></td>
<td>- FAILED</td>
</tr>
<tr>
<td>failedSourceRows</td>
<td>Number of rows that were not read from the source.</td>
</tr>
<tr>
<td>successSourceRows</td>
<td>Number of rows that were successfully read from the source.</td>
</tr>
<tr>
<td>failedTargetRows</td>
<td>Number of rows that were not written to the target.</td>
</tr>
<tr>
<td>successTargetRows</td>
<td>Number of rows that were successfully written to the target.</td>
</tr>
<tr>
<td>errorMsg</td>
<td>Error message associated with the job.</td>
</tr>
<tr>
<td>entries</td>
<td>Indicates the start of information for a child object.</td>
</tr>
<tr>
<td></td>
<td>- A child object might be a task within a task flow,</td>
</tr>
<tr>
<td></td>
<td>- or an object in a data replication task or D&amp;B360 workflow.</td>
</tr>
</tbody>
</table>

**Example**

To request information about tasks and task flows from the activity monitor, you might use the following URL and parameters:

https://app.informaticaondemand.com/saas/api/1/activitylog

Parameters:

- icSessionId=r21WJdXxrfKgTEcKgGc9Duc65hn2k2ggJDgIRjCB7XX5vArsjLS2xj19NQK5WECf
- details=false

**dnbbatch**

Use this resource to view or update the details of a D&B360 batch. For Informatica Cloud D&B360 partners only.

**GET Syntax**

To view the details of a D&B360 batch, use the following syntax.

```
 dnbbatch?icSessionId=<IC session ID>&responseType=<xml|json>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>icSessionId</td>
<td>Required</td>
<td>Informatica Cloud REST API session ID.</td>
</tr>
<tr>
<td>responseType</td>
<td>Optional</td>
<td>Format for the response from Informatica Cloud. Use one of the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- json. Returns information in JSON.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- xml. Returns information in XML.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- By default, Informatica Cloud responds in XML.</td>
</tr>
</tbody>
</table>
Return

When you request details about the D&B360 batch, Informatica Cloud returns the following information:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sfUsername</td>
<td>Salesforce user name.</td>
</tr>
<tr>
<td>sfPassword</td>
<td>Salesforce password.</td>
</tr>
<tr>
<td>sfSecurityToken</td>
<td>Salesforce security token.</td>
</tr>
<tr>
<td>sfServiceUrl</td>
<td>Salesforce URL.</td>
</tr>
<tr>
<td>rowLimit</td>
<td>Batch row limit.</td>
</tr>
<tr>
<td>enableNewBatch</td>
<td>Whether a new batch is enabled.</td>
</tr>
<tr>
<td>newBatchStartTime</td>
<td>New batch start time.</td>
</tr>
<tr>
<td>newBatchRepeatFrequency</td>
<td>New batch repeat frequency.</td>
</tr>
<tr>
<td>newBatchRepeatInterval</td>
<td>New batch repeat interval.</td>
</tr>
<tr>
<td>enableRefreshBatch</td>
<td>Whether a refresh batch is enabled.</td>
</tr>
<tr>
<td>refreshBatchStartTime</td>
<td>Refresh batch start time.</td>
</tr>
<tr>
<td>refreshBatchRepeatFrequency</td>
<td>Refresh batch repeat frequency.</td>
</tr>
<tr>
<td>refreshBatchRepeatInterval</td>
<td>Refresh batch repeat interval.</td>
</tr>
</tbody>
</table>

POST Syntax

To update the details of a D&B360 batch, use the following syntax:

```
dnbbatch?icSessionId=<IC session ID>&sfUsername=<Salesforce user name>&sfPassword=<Salesforce password>&sfSecurityToken=<Salesforce security token>&rowLimit=<row limit>&enableNewBatch=<new batch enabled>&newBatchStartTime=<new batch start time>&newBatchRepeatFrequency=<new batch repeat frequency>&newBatchRepeatInterval=<new batch repeat interval>&enableRefreshBatch=<refresh batch enabled>&refreshBatchStartTime=<refresh batch start time>&refreshBatchRepeatFrequency=<refresh batch repeat frequency>&refreshBatchRepeatInterval=<refresh batch repeat interval>&responseType=<xml|json>
```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>newBatchStartTime</td>
<td>Optional</td>
<td>New batch start time. Required if a new batch is enabled.</td>
</tr>
<tr>
<td>newBatchRepeatFrequency</td>
<td>Optional</td>
<td>New batch repeat frequency. Required if a new batch is enabled.</td>
</tr>
<tr>
<td>newBatchRepeatInterval</td>
<td>Optional</td>
<td>New batch repeat interval. Required if a new batch is enabled.</td>
</tr>
<tr>
<td>enableRefreshBatch</td>
<td>Optional</td>
<td>Whether a refresh batch is enabled.</td>
</tr>
<tr>
<td>refreshBatchStartTime</td>
<td>Optional</td>
<td>Refresh batch start time. Required if a refresh batch is enabled.</td>
</tr>
<tr>
<td>refreshBatchRepeatFrequency</td>
<td>Optional</td>
<td>Refresh batch repeat frequency. Required if a refresh batch is enabled.</td>
</tr>
<tr>
<td>refreshBatchRepeatInterval</td>
<td>Optional</td>
<td>Refresh batch repeat interval. Required if a refresh batch is enabled.</td>
</tr>
<tr>
<td>responseType</td>
<td>Optional</td>
<td>Format for the response from Informatica Cloud. Use one of the following options: - json. Returns information in JSON. - xml. Returns information in XML. By default, Informatica Cloud responds in XML.</td>
</tr>
</tbody>
</table>

**Return**

When you update the D&B360 batch, Informatica Cloud returns a code that indicates the success or failure of the update.

**Example**

To view information about a D&B360 batch, you might use the following URL and parameter:

https://app.informaticaondemand.com/saas/api/1/dnbbatch

Parameter:

icSessionId=r2WJdXxrfKgTEcKgGc9Duc65hnZk2ggJdgIRjCB7XX5wAsjLZxJ1NQK5WEcf

**login**

Use this resource to log in to Informatica Cloud and create an Informatica Cloud REST API session. Returns a session ID that you can use to perform additional API calls without additional login requirements. Resources used to monitor jobs require a session ID.
POST Syntax

```
login?username=<IC username>&password=<IC password>&responseType=<xml|json>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>Required</td>
<td>Informatica Cloud user name.</td>
</tr>
<tr>
<td>password</td>
<td>Required</td>
<td>Informatica Cloud password.</td>
</tr>
<tr>
<td>responseType</td>
<td>Optional</td>
<td>Format for the response from Informatica Cloud. Use one of the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- json. Returns information in JSON.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- xml. Returns information in XML.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>By default, Informatica Cloud responds in XML.</td>
</tr>
</tbody>
</table>

OR:

```
login?sfSessionId=<Salesforce Session ID>&sfServerUrl=<Salesforce URL>&responseType=<xml|json>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sfSessionId</td>
<td>Required</td>
<td>Salesforce session ID.</td>
</tr>
<tr>
<td>sfServerUrl</td>
<td>Required</td>
<td>Salesforce URL.</td>
</tr>
<tr>
<td>responseType</td>
<td>Optional</td>
<td>Format for the response from Informatica Cloud. Use one of the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- json. Returns information in JSON.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- xml. Returns information in XML.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>By default, Informatica Cloud responds in XML.</td>
</tr>
</tbody>
</table>

Return

An Informatica Cloud REST API session ID that you can use for additional API calls, such as the following:

```
iJJePC5K+i1ntsaNV8kGI1MCD13QBr8rup4grRT0Hnyzhim2FkuP7qRX2Iy1g8MY
```

The following is a sample return in XML:

```
<?xml version="1.0" encoding="utf-8"?>
<LoginResponse>
  <errorCode>NO_ERROR</errorCode>
  <errorMsg></errorMsg>
  <sessionId>iJJePC5K+i1ntsaNV8kGI1MCD13QBr8rup4grRT0Hnyzhim2FkuP7qRX2Iy1g8MY</sessionId>
</LoginResponse>
```

Example

To create a session ID using his Informatica Cloud user name and requesting the response in JSON, John Doe might use the following URL and parameters:

```
https://app.informaticaondemand.com/saas/api/1/login
```

Parameters:

- username=jdoe@xyz.com
- password=jdoepassword
- responseType=json

To create a session ID using a Salesforce session ID, you might use the following URL and parameters:

```
https://app.informaticaondemand.com/saas/api/1/login
```

Parameters:
Use this resource to create an Informatica Cloud organization. For Informatica Cloud partners only.

**POST Syntax**

```
register?offerCode=<offer code>&sfSessionId=<Salesforce session ID>&sfServerUrl=<Salesforce URL>&responseType=<xml|json>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>offerCode</td>
<td>Required</td>
<td>Offer code assigned to Informatica Cloud partners.</td>
</tr>
<tr>
<td>sfSessionId</td>
<td>Required</td>
<td>Salesforce session ID.</td>
</tr>
<tr>
<td>sfServerUrl</td>
<td>Required</td>
<td>Salesforce URL.</td>
</tr>
<tr>
<td>responseType</td>
<td>Optional</td>
<td>Format for the response from Informatica Cloud. Use one of the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- json. Returns information in JSON.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- xml. Returns information in XML. By default, Informatica Cloud responds in XML.</td>
</tr>
</tbody>
</table>

**OR:**

```
register?offerCode=<offer code>&orgName=organization name&firstName=first name&lastName=last name&title=title&email=email address&phone=phone number&address=address&city=city&state=state&zipCode=zip code&country=country&employees=employees&responseType=<xml|json>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>offerCode</td>
<td>Required</td>
<td>Offer code assigned to Informatica Cloud partners.</td>
</tr>
<tr>
<td>orgName</td>
<td>Required</td>
<td>Name for the new Informatica Cloud organization.</td>
</tr>
<tr>
<td>firstName</td>
<td>Required</td>
<td>First name of the organization administrator.</td>
</tr>
<tr>
<td>lastName</td>
<td>Required</td>
<td>Last name of the organization administrator.</td>
</tr>
<tr>
<td>title</td>
<td>Optional</td>
<td>Title of the organization administrator.</td>
</tr>
<tr>
<td>email</td>
<td>Required</td>
<td>Email address for the organization administrator.</td>
</tr>
<tr>
<td>phone</td>
<td>Optional</td>
<td>Phone number for the organization administrator.</td>
</tr>
<tr>
<td>address</td>
<td>Optional</td>
<td>Address where the organization is located.</td>
</tr>
<tr>
<td>city</td>
<td>Optional</td>
<td>City where the organization is located.</td>
</tr>
<tr>
<td>state</td>
<td>Optional</td>
<td>State where the organization is located.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Required/Optional</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>zipcode</td>
<td>Optional</td>
<td>Zip code where the organization is located.</td>
</tr>
<tr>
<td>country</td>
<td>Optional</td>
<td>Country where the organization is located.</td>
</tr>
<tr>
<td>employees</td>
<td>Optional</td>
<td>Number of employees in the organization.</td>
</tr>
<tr>
<td>responseType</td>
<td>Optional</td>
<td>Format for the response from Informatica Cloud. Use one of the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- json. Returns information in JSON.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- xml. Returns information in XML.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>By default, Informatica Cloud responds in XML.</td>
</tr>
</tbody>
</table>

**Return**

An Informatica Cloud organization ID, such as the following:

```
org1235
```

**Example**

To create an organization using an offer code and Salesforce session ID, you might use the following URL and parameters:

```
https://app.informaticaondemand.com/saas/api/1/register
```

Parameters:

- offerCode=35434
- sfSessionID=34352
- sfServerURL=https://www.salesforce.com/services/Soap/u/19.0

---

**runjob**

Use this resource to start an Informatica Cloud task or task flow.

**POST Syntax**

```
runjob?icSessionId=<IC session ID>=&jobName=<task or taskflow name>&jobType=<task type>&responseType=<xml|json>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>icSessionId</td>
<td>Required</td>
<td>Informatica Cloud REST API session ID.</td>
</tr>
<tr>
<td>jobName</td>
<td>Required</td>
<td>Task or taskflow name.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Required/Optional</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| jobType   | Required         | Type of task or taskflow. Use one of the following options:  
- DLS. Data Loader task.  
- DSS. Data Synchronization task.  
- DRS. Data Replication task.  
- DQA. Data Assessment task.  
- PCS. PowerCenter task.  
- DNB. D&B360 batch.  
- Workflow. Task flow. |
| responseType | Optional | Format for the response from Informatica Cloud. Use one of the following options:  
- json. Returns information in JSON.  
- xml. Returns information in XML.  
By default, Informatica Cloud responds in XML. |

OR:

`runjob?username=<IC user name>&password=<IC password>&jobName=<task or taskflow name>&jobType=<task type>&responseType=<xml|json>`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>Required</td>
<td>Informatica Cloud user name.</td>
</tr>
<tr>
<td>password</td>
<td>Required</td>
<td>Informatica Cloud password.</td>
</tr>
<tr>
<td>jobName</td>
<td>Required</td>
<td>Task or taskflow name.</td>
</tr>
</tbody>
</table>
| jobType   | Required         | Type of task or taskflow. Use one of the following options:  
- DLS. Data Loader task.  
- DSS. Data Synchronization task.  
- DRS. Data Replication task.  
- DQA. Data Assessment task.  
- PCS. PowerCenter task.  
- DNB. D&B360 batch.  
- Workflow. Task flow. |
| responseType | Optional | Format for the response from Informatica Cloud. Use one of the following options:  
- json. Returns information in JSON.  
- xml. Returns information in XML.  
By default, Informatica Cloud responds in XML. |

**Return**

A code indicating the success or failure of the request:

- true. The request was successful.
- false. The request failed.

The following is a sample response in XML:

```xml
<?xml version="1.0" encoding="utf-8"?>
<RunJobResponse>
  <errorCode>NO_ERROR</errorCode>
  <errorMsg></errorMsg>
  <success>true</success>
</RunJobResponse>
```
Example
To start a data replication task called Replicate_All, you might use the following URL and parameters:

https://app.informaticaondemand.com/saas/api/1/runjob

Parameters:
icSessionId=r21WJdXxrFkJGeKgGc9Duc65hznk2ggJdgI8jCB7XX5vAsjLSZxJ19NQK5WEEc
jobName=Replicate_All
jobType=DRS

salesforceversion

Use this resource to return the Salesforce version used by Informatica Cloud.

GET Syntax

salesforceversion?responseType=<xml|json>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>responseType</td>
<td>Optional</td>
<td>Format for the response from Informatica Cloud. Use one of the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- json. Returns information in JSON.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- xml. Returns information in XML. By default, Informatica Cloud responds in XML.</td>
</tr>
</tbody>
</table>

Return

The Salesforce version used by Informatica Cloud.

The following is a sample return in XML:

```xml
<?xml version="1.0" encoding="utf-8"?>
<SalesforceVersionResponse>
<errorCode>NO_ERROR</errorCode>
<errorMsg></errorMsg>
<versionNo>19</versionNo>
</SalesforceVersionResponse>
```

Example

To verify the Salesforce version used by Informatica Cloud, you can use the following URL:

https://app.informaticaondemand.com/saas/api/1/salesforceversion

servertime

Use this resource to return the local time for the Informatica Cloud server.
GET Syntax

server?time?responseType=<xml|json>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
</table>
| responseType | Optional          | Format for the response from Informatica Cloud. Use one of the following options:
|             |                   | - json. Returns information in JSON.
|             |                   | - xml. Returns information in XML. |
|             |                   | By default, Informatica Cloud responds in XML. |

Return

The local time for the Informatica Cloud server.

Example

To verify the local time for the Informatica Cloud server, you can use the following URL:

https://app.informaticaondemand.com/saas/api/1/server?time

user

Use this resource to view or update the details of an Informatica Cloud user account. You can also use this resource to create an Informatica Cloud user account. To ensure organization security, this resource does not display or update the password for a user account.

GET Syntax

To view the details of an Informatica Cloud user account, use the following syntax.

user?icSessionId=<IC session ID>&username=<IC user name>&responseType=<xml|json>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>icSessionId</td>
<td>Required</td>
<td>Informatica Cloud REST API session ID.</td>
</tr>
<tr>
<td>username</td>
<td>Required</td>
<td>Informatica Cloud user name.</td>
</tr>
</tbody>
</table>
| responseType | Optional          | Format for the response from Informatica Cloud. Use one of the following options:
|             |                   | - json. Returns information in JSON.
|             |                   | - xml. Returns information in XML. |
|             |                   | By default, Informatica Cloud responds in XML. |
Return

When you request details about a user account, Informatica Cloud returns the following information:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>Informatica Cloud user name.</td>
</tr>
<tr>
<td>sfUsername</td>
<td>Salesforce user name.</td>
</tr>
<tr>
<td>firstName</td>
<td>First name for the user account.</td>
</tr>
<tr>
<td>lastName</td>
<td>Last name for the user account.</td>
</tr>
<tr>
<td>title</td>
<td>Title of the user.</td>
</tr>
<tr>
<td>phone</td>
<td>Phone number for the user.</td>
</tr>
<tr>
<td>description</td>
<td>Description of the user.</td>
</tr>
<tr>
<td>createTime</td>
<td>When the user account was created.</td>
</tr>
<tr>
<td>updateTime</td>
<td>When the user account was last updated.</td>
</tr>
<tr>
<td>createdBy</td>
<td>Informatica Cloud user who created the user account.</td>
</tr>
<tr>
<td>updatedBy</td>
<td>Informatica Cloud user who last updated the user account.</td>
</tr>
</tbody>
</table>

POST Syntax

To create an Informatica Cloud user account or update the details of an existing account, use the following syntax:

```
user?icSessionId=<IC session ID>&username=<IC user name>&sfUsername=<Salesforce user name>&firstName=<first name>&lastName=<last name>&title=<title>&phone=<phone>&description=<description>&responseType=<xml|json>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required/ Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>icSessionId</td>
<td>Required</td>
<td>Informatica Cloud REST API session ID.</td>
</tr>
<tr>
<td>username</td>
<td>Required</td>
<td>Informatica Cloud user name.</td>
</tr>
<tr>
<td>sfUsername</td>
<td>Optional</td>
<td>Salesforce user name.</td>
</tr>
<tr>
<td>firstName</td>
<td>Optional</td>
<td>First name for the user account.</td>
</tr>
<tr>
<td>lastName</td>
<td>Optional</td>
<td>Last name for the user account.</td>
</tr>
<tr>
<td>title</td>
<td>Optional</td>
<td>Title of the user.</td>
</tr>
<tr>
<td>phone</td>
<td>Optional</td>
<td>Phone number for the user.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Required/Optional</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>description</td>
<td>Optional</td>
<td>Description of the user.</td>
</tr>
<tr>
<td>responseType</td>
<td>Optional</td>
<td>Format for the response from Informatica Cloud. Use one of the following options: - json. Returns information in JSON. - xml. Returns information in XML. By default, Informatica Cloud responds in XML.</td>
</tr>
</tbody>
</table>

**Return**

When you create or update a user account, Informatica Cloud returns a code that indicates the success or failure of the update.

**Example**

To view information about the John Doe user account, you might use the following URL and parameters:

https://app.informaticaondemand.com/saas/api/1/user

Parameters:

icSessionId=r21WJdXxrKgTEcKgc9Duc55hn5k2ggJdgRjCB7XX5wAsjLSZxJ19NQK5MEECf
username=jdoe@xyz.com
INDEX

A
ABORT function
description 207
ABS function
description 208
absolute values
obtaining 208
access permissions See Permissions
active plug-ins
description 60
activity log
description 158
searching based on activity log details 161
searching based on task name 161
viewing jobs 158
activity log details
viewing jobs 159
activity monitor
description 162
detail view 162
list view 162
stopping a job 162
viewing 162
activitylog
REST API resource 324
activitymonitor
REST API resource 326
ADD_TO_DATE function
description 209
Address Validation
quality metric 68
Admin role
description 16
advanced data filters
configuring 146
Advanced Encryption Standard algorithm
description 211
AES_DECRYPT function
description 210
AES_ENCRYPT function
description 211
AND
reserved word 144
API
Informatica Cloud REST API 318
arithmetic
date/time values 312
arithmetic operators
description 297
using strings in expressions 297
using to convert data 297
ASCII
CHR function 214
ASCII function
description 212
assignment rule selection
Salesforce targets in data synchronization tasks 127
audit log
description 25

B
bigint
converting values to 283
billing information
editing 25
viewing 25
browser
requirements 3

calendars
date types supported 303
capitalization
strings 235, 245, 295
case
converting to uppercase 295
CEIL function
description 213
changing organizations
in the Informatica Cloud Administrator 27
character functions
ASCII 212
CHR 214
CHR_CODE 215
CONCAT function 216
INITCAP 235
INSTR 236
LENGTH 243
list of 195
LOWER 245
LPAD 246
LTRIM 247
METAPHONE 250
REG_EXTRACT 259
REG_MATCH 261
REG_REPLACE 262
REPLACECHR 263
REPLACESTR 265
RPAD 271
RTRIM 272
SOUNDEX 277
SUBSTR 279
UPPER 295
character strings
converting from dates 284
converting to dates 288
Index
database sources and targets
rules in data synchronization tasks 111
database targets
configuring update columns for data synchronization tasks 127
database user accounts
requirements for data loader 84
requirements for data synchronization 109
user privileges for data replication 93
databases
connections 41
dates 304
rules for sources and targets in data loader tasks 83
datatypes
configuring 113
conversion 177
conversion from Microsoft SQL Server 2000 179
conversion from Microsoft SQL Server 2005 180
conversion from Microsoft SQL Server 2008 182
conversion from MySQL 183
conversion from Oracle 185
conversion from Salesforce to Microsoft SQL Server 187
conversion from Salesforce to MySQL 188
conversion from Salesforce to Oracle 190
Date/Time 302
MySQL 188
rules and guidelines 177
source and target inconsistencies 106
date functions
ADD_TO_DATE 209
DATE_COMPARE 221
DATE_DIFF 221
GET_DATE_PART 229
LAST_DAY 242
MAKE_DATE_TIME 248
ROUND 268
SET_DATE_PART 274
SYSTIMESTAMP 281
TRUNC (Dates) 292
DATE_COMPARE function
description 221
DATE_DIFF function
description 221
date/time values
adding 209
dates
converting to character strings 284
databases 304
default format 305
flat files 305
format strings 305
functions 197
Julian 303
Modified Julian 303
overview 302
performing arithmetic 312
rounding 268
truncating 292
year 2000 303
Daylight Savings Time
schedules 55
DD_DELETE constant
reserved word 144
DD_INSERT constant
reserved word 144
DD_REJECT constant
reserved word 144
DD_UPDATE constant
reserved word 144
DEC_BASE64 function
description 223
decimal values
converting 290
DECODE function
description 224
decoding
DEC_BASE64 function 223
DECOMPRESS function
description 225
decryption
AES_DECRYPT function 210
default date format
defined 305
default values
ERROR function 226
delete
task operations 47
delete criteria
description 96
Designer role
description 16
detail view
activity monitor 162
directories
configuring Secure Agent login to access 36
division calculation
returning remainder 253
dnbbatch
REST API resource 327
double precision values
floating point numbers 290
downloading
data synchronization task mapping XML 131
Duplicates
customizing weights 76
quality metric 68
E
e-mail
configuring for data assessment tasks 75
configuring for data replication tasks 98
configuring for data synchronization tasks 127
configuring for PowerCenter tasks 135
e-mail addresses
for password reset 6
e-mail notification
configuring for task flows 58
configuring for tasks 149
configuring for the organization 11
overview 10
empty strings
testing for 243
ENC_BASE64 function
description 226
encoding
description 226
encoding functions
AES_DECRYPT 210
AES_ENCRYPT 211
COMPRESS 216
CRC32 219
DEC_BASE64 223
DECOMPRESS 225
description 198
encryption

AES_ENCRYPT function 211
using the Advanced Encryption Standard algorithm 211

ERROR function

default value 226
description 226

temperature file
viewing snapshot 164
temperature log files
local directory 163
exception handling
in plug-in stored procedures 61
in stored procedures for PowerCenter tasks 133
excluding fields
from data replication tasks 97

EXP function

description 227
exponent values
calculating 227
returning 256
expressions
rules 143

:EXT reference qualifier
reserved word 144

external IDs
example 140
for related objects 139

F

FALSE constant

description 300
reserved word 144
field exclusions
configuring for data replication tasks 96
field expressions
comments, adding 143
components 142
conditional 300
creating 141
literals 142
null constraints 300
reserved words 144
rules 143
rules for validating 141
syntax 142
using operators 296
field mappings
configuring for data loader task 87
configuring for data synchronization tasks 125
customizing 77
guidelines for customization 77
required fields 112
troubleshooting 175
understanding for Data Synchronization tasks 112
field scores

data assessment tasks 70
fields
required in field mappings 112
filter conditions
null values 301
financial functions
description 198
FV function 228
NPV function 254
PMT function 255

PV function 257
RATE function 258

Flat File connections
configuring details 47
rules and guidelines 44

flat files
connection 41
creating target files in data synchronization tasks 110
dates 305
directory for data loader tasks 84
directory for data replication tasks 93
directory for data synchronization tasks 109
rules for sources and targets in data loader tasks 83
rules for sources and targets in data synchronization tasks 111

FLOOR function
description 228

format
from character string to date 288
from date to character string 284

format strings
dates 305
definition 302
Julian day 306
matching 311
Modified Julian day 306
TO_CHAR function 306
TO_DATE and IS_DATE format strings 309

free subscription license
description 13

FTP/SFTP
connection 41
FTP/SFTP connections
configuring details 47
for PowerCenter tasks 42
overview 42
rules and guidelines 44

full load
and non-replicable objects 106
data replication load type 106

functions
categories 195
character 195
conversion 196
data cleansing 196
date 197
definition 142
encoding 198
financial 198
numeric 198
scientific 199
special 199
string 199
test 199

FV function
description 228

G

GET_DATE_PART function
description 229

GREATEST function
description 231

Gregorian calendar
in date functions 303
installation requirements 3
INSTR function
description 236
integers
converting other values 291
IS_DATE function
description 238
format strings 309
IS_NUMBER function
description 239
IS_SPACES function
description 240
ISNULL function
description 241

J
J format string
using with IS_DATE 311
using with TO_CHAR 308
using with TO_DATE 311
jobs
monitoring 158
Julian dates
in date functions 303
Julian day
format string 306, 309

L
LAST_DAY function
description 242
LEAST function
description 243
LENGTH function
description 243
empty string test 243
license expiration messages
ignoring 6
license information
viewing 13
license types
overview 13
licenses
expiration 4
Linux
configuring proxy settings 40
Secure Agent 37
Secure Agent installation 37
starting and stopping the Secure Agent 39
uninstalling the Secure Agent 40
list view
activity monitor 162
literals
definition 142
single quotation mark requirement 142
single quotation marks in 214, 216
string and numeric 142
::LKP reference qualifier
reserved word 144
LN function
description 244
load types
full load 106
incremental load 106
log files
error 163
LOG function
description 245
logarithm
returning 244, 245
logging in
Informatica Cloud Administrator 27
logical operators
description 299
login
description 3
lock out 4
non-administrator user 4
organization administrator 4
REST API resource 330
login denied
troubleshooting 176
lookups
condition 114
rules for creating 114
LOWER function
description 245
LPAD function
description 246
LTRIM function
description 247
Minimum Character Mix
organization property 10
MOD function
description 253
Modified Julian day
format string 306, 309
monitoring
jobs 158
month
returning last day 242
Monthly Job Row Count
verifying processed rows for the month 6
MS Access
connection 41
MS Access connections
configuring details 47
rules and guidelines 45
multiple searches
example of TRUE constant 301
multiple sources
rules 110
MySQL
column names 103
datatype conversion 183
datatypes 188
table names 103
MySQL connections
configuring details 47
MySQL database
connection 41
N
negative values
SIGN 275
nested expressions
operators 296
non-replicateable objects
loading 106
NOT
reserved word 144
NPER function
description 254
NULL constant
description 300
reserved word 144
null updates
configuring for Salesforce targets in data synchronization tasks 127
null values
checking for 241
filter conditions 301
logical operators 299
operators 301
string operator 298
numbers
rounding 270
truncating 294
numeric functions
ABS 208
CEIL 213
CONVERT_BASE 217
CUME 220
description 198
EXP 227
FLOOR 228
LN 244
LOG 245
organization administrator
  overview 3
organization details
  configuring 11
  editing 10
organization hierarchy
  creating a sub-organization 28
  overview 27
  using the Informatica Cloud Administrator 27
organization properties
  viewing 8
organization properties and details
  configuring 11
organization security
  overview 14

P
password complexity
  configuring for the organization 11
passwords
  complexity options 10
  overview 4
perl compatible regular expression syntax
  using in a REG_EXTRACT function 259
  using in a REG_MATCH function 259
permissions
  configuring for a user group 17
  configuring for individual objects 19
  for individual objects 19
  required to perform tasks 15
plug-ins
  active and passive 60
  configuring 62
  deleting 63
  in data synchronization tasks 113
  overview 60
  rules for PowerCenter XML files 61
  rules for using in tasks 62
  viewing 63
PMT function
  description 255
positive values
  SIGN 275
postprocessing commands
  SQL commands 98, 115
POWER function
  description 256
PowerCenter service
  description 132
PowerCenter tasks
  configuring 135
  deleting 138
  monitoring 138
  prerequisites 132
  rules for configuring 135
  rules for running 137
  rules for sources and targets 134
  running 137
  running manually 137
  running on a schedule 138
  stopping 138
  supported transformations and mapping objects 133
  troubleshooting 174
  viewing details 137
PowerCenter XML files
  rules for plug-ins 61
supported transformations and mapping objects 133
precision
source and target inconsistencies 106
preprocessing and postprocessing commands
classifying for data replication tasks 98
classifying for data synchronization tasks 127
rules for operating system commands 98, 116
rules for SQL commands 98, 115
preprocessing commands
SQL commands 98, 115
prerequisites
data replication 93
for data loader tasks 84
for data synchronization tasks 109
for PowerCenter tasks 132
primary key constraint
null values 300
privileges
description 16
PROC_RESULT variable
reserved word 144
proxy settings
configuring on Linux 40
configuring on Windows 35
PV function
description 257

Q
quality metrics
Address Validation 68
Completeness 68
classifying 73
Conformance 68
description 68
Duplicates 68
types 68
quick reference
REST API 322
Quick Setup page
description 5
quotation marks
inserting single using CHR function 142

R
RAND function
description 258
RATE function
description 258
REG_EXTRACT function
description 259
using perl syntax 259
REG_MATCH function
description 261
using perl syntax 259
REG_REPLACE function
description 262
register
REST API resource 331
registering
Secure Agent on Linux 37
related objects
configuring external IDs 139
example 140
repeat frequency
description 55
for schedules 54
REPLACECHR function
description 263
REPLACESTR function
description 265
requirements
for data assessment sources 72
for migration 24
incremental load 106
installation 3
reserved words
list 144
resetting
target tables 105
response codes
REST API 321
response formats
REST API 319
REST API
documentation conventions 319
guidelines 320
overview 318
quick reference 322
response formats 319
return codes 321
test page 319
XML schema definition (XSD) 319
restrictions
subscription options 2
REVERSE function
description 267
roles
best practices 15
overview 16
ROUND (dates) function
description 268
ROUND (numbers) function
description 270
rounding
dates 268
numbers 270
rows
avoiding spaces 240
running total 220
skipping 226
RPAD function
description 271
RR format string
description 303
difference between YY and RR 304
using with IS_DATE 311
using with TO_CHAR 309
using with TO_DATE 311
RTRIM function
description 272
runjob
REST API resource 332
running total
returning 220

S
Salesforce
configuring the security token or trusted IP ranges 43
connections 41
datatype changes 104
datatype conversion to Microsoft SQL Server 187
| Index | 347 |

| Datatype conversion to MySQL | 188 |
| Datatype conversion to Oracle | 190 |
| Including deleted or archived source data | 92 |
| Including historical data in data synchronization tasks | 109 |
| Rules and guidelines | 46 |
| Rules for sources and targets in data loader tasks | 83 |
| Rules for sources and targets in data synchronization tasks | 112 |
| Target batch size for data synchronization tasks | 127 |

Salesforce connections
- Configuring details | 47 |
- Overview | 43 |
Salesforce IDs
- For objects | 139 |
- Understanding | 116 |
Salesforce targets
- Assignment rule selection for data synchronization tasks | 127 |
- Configuring null updates for data synchronization tasks | 116 |
- SalesforceVersion
  - REST API resource | 334 |
SAP
- Connections | 41 |
SAP connections
- Configuring details | 47 |
- Overview | 43 |
scale
- Source and target inconsistencies | 106 |
schedule
- Configuring for a PowerCenter task | 135 |
schedule offset
- Organization property | 10 |
schedules
- Associating with data synchronization tasks | 130 |
  - Configuring | 55 |
  - Configuring for data loader tasks | 88 |
  - Configuring for data replication tasks | 98 |
  - Configuring for data synchronization tasks | 127 |
  - Data assessment tasks, associating with | 79 |
  - Data replication tasks, associating with | 101 |
  - Data synchronization task | 115 |
Daylight Savings Time | 55 |
deleting | 58 |
description | 54 |
for data assessment tasks | 79 |
for data loader tasks | 90 |
for PowerCenter tasks | 138 |
repeat frequency | 54 |
task flows, associating with | 60 |
time zones | 55 |
viewing details | 57 |
scientific functions
- COS | 218 |
- COSH | 218 |
  - Description | 199 |
  - SIN | 276 |
  - SINH | 277 |
  - TAN | 282 |
  - TANH | 282 |
scorecard thresholds
- For data assessment tasks | 75 |
scorecards
- Adding to dashboard | 75 |
  - Description | 70 |
  - Thresholds, configuring | 75 |
  - Viewing | 79 |

:SD reference qualifier
reserved word | 144 |
Secure Agent
- Configuring a Windows service login | 36 |
  - Deleting | 36 |
  - Description | 1 |
  - Downloading | 33 |
  - For data loader | 84 |
  - For data synchronization | 109 |
  - For PowerCenter tasks | 132 |
  - Installing and registering on Linux | 37 |
  - Installing on Windows | 33 |
  - On Linux | 37 |
  - On Windows | 32 |
  - Overview | 31 |
  - Requirements | 3 |
  - Rules and guidelines | 32 |
  - Starting and stopping on Linux | 39 |
  - Stopping and restarting on Windows | 35 |
  - Troubleshooting | 165 |
  - Uninstalling on Linux | 40 |
  - Uninstalling on Windows | 36 |
  - View details, refresh status | 34 |
  - Viewing history | 34 |
  - Viewing status | 34 |
  - Viewing status from Secure Agent Manager | 34 |
  - Viewing status on Linux | 38 |
Secure Agent Manager
- Secure Agent status | 34 |
- Stopping and restarting the Secure Agent | 35 |
  - Understanding | 32 |
security
- Informatica Cloud | 2 |
  - Rules and guidelines | 15 |
  - Troubleshooting | 176 |
  - Two-factor authentication | 11 |
security token
- Configuring | 43 |
:SEQ reference qualifier
reserved word | 144 |
server time
- REST API resource | 335 |
service name
- Description | 47 |
- Oracle | 47 |
service URL
- Overview | 46 |
Salesforce | 47 |
session commands
- PowerCenter tasks
  - Using pre- and post-session commands | 134 |
sessions
- Stopping | 207 |
SESSIDSTARTTIME variable
reserved word | 144 |
using in date functions | 312 |
SET_DATE_PART function
- Description | 274 |
SID
- Oracle | 47 |
SIGN function
- Description | 275 |
simple data filters
- Configuring | 145 |
SIN function
- Description | 276 |
sine
- Returning | 276, 277 |
single quotation marks in string literals
CHR function 214
using CHR and CONCAT functions 216
SINH function
description 277
skipping
rows 226
SOUNDEX function
description 277
sources
configuring for data loader task 85
configuring for data replication tasks 94
configuring for data synchronization task 118
configuring multiple-object sources 121
configuring single-object sources 118, 119
custom 63
for data assessment tasks 72
rules for multiple-object sources in data synchronization tasks 110
:SP reference qualifier
reserved word 144
spaces
avoiding in rows 240
special functions
ABORT 207
DECODE 224
description 199
ERROR 226
IIF 231
SPOUTPUT
reserved word 144
SQL commands
rules for data replication tasks 98, 115
rules for data synchronization tasks 98, 115
SQL IS_CHAR function
using REG_MATCH 261
SQL LIKE function
using REG_MATCH 261
SQL Server
connection 41
SQL syntax
converting to perl syntax 259
SQRT function
description 279
square root
returning 279
SSL
security 2
SSSSS format string
using with IS_DATE 312
using with TO_CHAR 308
using with TO_DATE 312
status
description 158
Stop Job
activity monitor option 162
stopping
sessions 207
stored procedures
exception handling for PowerCenter tasks 133
exception handling in plug-ins 61
string conversion
dates 303
string functions
CHOOSE 213
description 199
INDEXOF 234
REVERSE 267
string literals
single quotation mark requirement 142
strings
adding blanks 246
adding characters 246
capitalization 235, 245, 295
character set 236
concatenating 216, 298
converting character strings to dates 288
converting dates to characters 284
converting length 271
converting numeric values to text strings 287
number of characters 243
removing blanks 247
removing blanks and characters 272
removing characters 247
replacing multiple characters 265
replacing one character 263
returning portion 279
sub-organizations
creating 28
in an organization hierarchy 27
managing with the Informatica Cloud Administrator 27
subscribe
Informatica Cloud Express 6
subscription license
description 13
subscription options
functional restrictions 2
subscriptions
description 5
SUBSTR function
description 279
success files
generating for Salesforce for data synchronization tasks 127
Salesforce 116
Support page
description 5
support requests
submitting 5
syntax
for field expressions 142
SYSDATE variable
reserved word 144
system requirements
minimum 3
SYSTIMESTAMP function
description 281

T

table names
duplicates 103
for Microsoft SQL Server target 103
for MySQL target 103
for Oracle target 103
truncating 170
tables
resetting 104
target database 102
user-defined indexes 105
TAN function
description 282
tangent
returning 282
functions
TANH function
description 282
target batch size
for Salesforce targets 116
Salesforce targets in data synchronization tasks 127
target databases
column names, truncating 103
for data replication 92
requirements for a data loader task 84
requirements for a data synchronization task 109
requirements for data replication 95
resetting tables 104
table and column names 102
target flat files
creating in data synchronization tasks 110
target prefix
description 96
guidelines 95
target tables
creating for data replication tasks 105
rules for resetting 105
targets
configuring for data loader tasks 85
configuring for data replication tasks 96
configuring for data synchronization tasks 122, 123
truncating in data synchronization tasks 110
task flows
configuring 58
deleting 60
description 58
running 59
running manually 59
running on a schedule 60
viewing details 59
task operations
delete 47
description 46
insert 46
update 46
upsert 47
tasks
configuring email notification 149
configuring field expressions 141
migrating 24
stopping a running job 162
:TD reference qualifier
reserved word 144
test functions
description 199
IS_DATE 238
IS_NUMBER 239
IS_SPACES 240
ISNULL 241
test page
REST API 319
testing
connections 53
text strings
converting numeric values 287
time zones
associating with a user 21
changing user preference 6
data filters 147
description 55
in email notification 10
TO_CHAR (dates) function
description 284
examples 308
format strings 306
TO_CHAR (numbers) function
description 287
TO_DATE function
description 288
examples 311
format strings 309
TO_DECIMAL function
description 290
TO_FLOAT function
description 290
TO_INTEGER function
description 291
transformations
in PowerCenter tasks 133
trial license
description 13
troubleshooting
connections 166
data assessment tasks 168
data loader tasks 168
data replication tasks 170
data synchronization tasks 172
field mappings 175
PowerCenter tasks 174
Secure Agent 165
security 176
TRUE constant
description 301
reserved word 144
TRUNC (dates) function
description 292
TRUNC (numbers) function
description 294
truncating
date/time values 303
dates 292
numbers 294
trust site
description xiv
trusted IP ranges
configuring 43
two-factor authentication
configuring for the organization 11
description 11
update
task operations 46
update columns
adding 116
configuring for database targets for data synchronization tasks 127
updates
boolean expressions 301
UPPER function
description 295
upsert
task operations 47
URL
Informatica Cloud Administrator 27
URLs
service URL 46
user
REST API resource 335
user accounts
creating 3

U
update
task operations 46
update columns
adding 116
configuring for database targets for data synchronization tasks 127
updates
boolean expressions 301
UPPER function
description 295
upsert
task operations 47
URL
Informatica Cloud Administrator 27
URLs
service URL 46
user
REST API resource 335
user accounts
creating 3
lockouts 4
user group overview 17
user groups
  best practices 15
  configuring 17
  deleting 19
  object-level permissions 19
  viewing details 19
user preferences
  description 6
  editing 6
  tasks 6
  viewing 6
users
  configuring 21
  deleting 23
  description 21
  preferences 6
  viewing details 23

V
validation
  rules 141
viewing
  Secure Agent status 38
  views See custom views

W
Web Service connection 41

Web Service connections
  configuring details 47
  overview 43
weights
  customizing 76
  guidelines for customization 76
Windows
  configuring proxy settings 35
Windows service
  configuring Secure Agent login 36
workflow XML file
  selecting for a PowerCenter task 135
WORKFLOWSTARTTIME variable
  reserved word 144

X
XML files
  rules for plug-ins 61
XML schema definition
  REST API 319
XSD See XML schema definition

Y
year 2000
  dates 303
YY format string
  difference between RR and YY 304
  using with TO_CHAR 309
  using with TO_DATE and IS_DATE 312