IBM InfoSphere Discovery
Version 4 Release 5.1

User Guide

IBM
### Chapter 3. Discovering data associations

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Map Horizontally Partitioned Target Tables</td>
<td>248</td>
</tr>
<tr>
<td>Perform Column Analysis</td>
<td>254</td>
</tr>
<tr>
<td>Overview</td>
<td>254</td>
</tr>
<tr>
<td>Perform Column Analysis</td>
<td>260</td>
</tr>
<tr>
<td>Re-run Column Analysis</td>
<td>261</td>
</tr>
<tr>
<td>Review Column Analysis Data</td>
<td>262</td>
</tr>
<tr>
<td>Change Data Type Formats</td>
<td>263</td>
</tr>
<tr>
<td>The Column Analysis Window</td>
<td>270</td>
</tr>
<tr>
<td>Discover PF Keys</td>
<td>284</td>
</tr>
<tr>
<td>Overview</td>
<td>284</td>
</tr>
<tr>
<td>Discover PF Keys</td>
<td>288</td>
</tr>
<tr>
<td>Re-Run PF Keys</td>
<td>291</td>
</tr>
<tr>
<td>Define PF Keys</td>
<td>293</td>
</tr>
<tr>
<td>Define Column Matches</td>
<td>303</td>
</tr>
<tr>
<td>Review PF Keys</td>
<td>313</td>
</tr>
<tr>
<td>The PF Keys Window</td>
<td>314</td>
</tr>
<tr>
<td>PF Keys Diagram</td>
<td>319</td>
</tr>
<tr>
<td>PF Keys Grid - Foreign Keys</td>
<td>332</td>
</tr>
<tr>
<td>PF Keys Grid - Column Matches</td>
<td>336</td>
</tr>
<tr>
<td>Discover Data Objects</td>
<td>338</td>
</tr>
<tr>
<td>Overview</td>
<td>338</td>
</tr>
<tr>
<td>Discover Data Objects</td>
<td>342</td>
</tr>
<tr>
<td>Re-run Data Objects</td>
<td>346</td>
</tr>
<tr>
<td>Define Data Objects</td>
<td>348</td>
</tr>
<tr>
<td>Create a Related Sample Set</td>
<td>356</td>
</tr>
<tr>
<td>Review Data Objects</td>
<td>358</td>
</tr>
<tr>
<td>The Data Objects Window</td>
<td>358</td>
</tr>
<tr>
<td>Data Objects List</td>
<td>359</td>
</tr>
<tr>
<td>Data Objects Diagram</td>
<td>361</td>
</tr>
<tr>
<td>Data Objects Grid</td>
<td>362</td>
</tr>
</tbody>
</table>

### Chapter 4. Discovering data transformations

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate Target Matches</td>
<td>473</td>
</tr>
<tr>
<td>Overview</td>
<td>473</td>
</tr>
<tr>
<td>Discover Target Matches</td>
<td>474</td>
</tr>
<tr>
<td>Re-run Target Matches</td>
<td>475</td>
</tr>
<tr>
<td>Defining Target Maps</td>
<td>477</td>
</tr>
<tr>
<td>Define a Target Match</td>
<td>480</td>
</tr>
<tr>
<td>Review Target Matches</td>
<td>487</td>
</tr>
<tr>
<td>Target Matches window</td>
<td>487</td>
</tr>
<tr>
<td>Target Matches List</td>
<td>488</td>
</tr>
<tr>
<td>Target Matches Diagram</td>
<td>489</td>
</tr>
<tr>
<td>Discover Maps</td>
<td>493</td>
</tr>
<tr>
<td>Overview</td>
<td>493</td>
</tr>
<tr>
<td>Discover Maps</td>
<td>494</td>
</tr>
<tr>
<td>Repeat Map Discovery</td>
<td>496</td>
</tr>
<tr>
<td>Defining a Map</td>
<td>496</td>
</tr>
<tr>
<td>Add a Source Table</td>
<td>499</td>
</tr>
<tr>
<td>Review Maps Summary</td>
<td>501</td>
</tr>
<tr>
<td>The Summary Window</td>
<td>502</td>
</tr>
<tr>
<td>Discover Joins</td>
<td>506</td>
</tr>
<tr>
<td>Overview</td>
<td>506</td>
</tr>
<tr>
<td>When to Run or Re-run Joins</td>
<td>507</td>
</tr>
<tr>
<td>Define a Join</td>
<td>510</td>
</tr>
<tr>
<td>Review Joins</td>
<td>512</td>
</tr>
<tr>
<td>The Joins Tab</td>
<td>512</td>
</tr>
<tr>
<td>The Join Condition Grid</td>
<td>513</td>
</tr>
<tr>
<td>Discover Binding Conditions</td>
<td>517</td>
</tr>
<tr>
<td>Overview</td>
<td>517</td>
</tr>
<tr>
<td>Generate Binding Conditions</td>
<td>519</td>
</tr>
<tr>
<td>Re-run Binding Conditions</td>
<td>521</td>
</tr>
<tr>
<td>Defining a Binding Condition</td>
<td>522</td>
</tr>
<tr>
<td>Review Binding Conditions</td>
<td>524</td>
</tr>
<tr>
<td>The Bindings Tab</td>
<td>524</td>
</tr>
<tr>
<td>The Binding Condition Grid</td>
<td>525</td>
</tr>
<tr>
<td>Binding Conditions Columns Grid</td>
<td>530</td>
</tr>
<tr>
<td>Discover Where Clause</td>
<td>532</td>
</tr>
<tr>
<td>Overview</td>
<td>533</td>
</tr>
<tr>
<td>Generate Where Clauses</td>
<td>533</td>
</tr>
<tr>
<td>Re-run Where Clauses</td>
<td>535</td>
</tr>
<tr>
<td>Defining a Where Clause</td>
<td>536</td>
</tr>
</tbody>
</table>
Chapter 1. Introducing IBM InfoSphere Discovery

IBM® InfoSphere® Discovery simplifies, streamlines, and accelerates enterprise-level data analysis and mapping projects. It automates the discovery of hidden sensitive data, undocumented business rules, data maps, and unknown data inconsistencies between data sets.

InfoSphere Discovery provides innovative data exploration and analysis techniques to automatically discover relationships and mappings among the structured data in the enterprise, based not on metadata but on actual values in the data itself. This means InfoSphere Discovery can detect relationships between tables and columns whose names or metadata alone do not suggest any connection. It also enables InfoSphere Discovery to identify and generate highly complex transformations that you can use to describe the locations and formats of sensitive data, describe the relationships of data elements across applications, or output as SQL or ETL code for use in later data transformation.

Although InfoSphere Discovery performs some phases of this mapping as automated processes, it also offers a broad set of powerful and flexible tools that analysts can use to manually review, develop, refine, and finally confirm a comprehensive set of data maps. These maps can then be used as blueprints in a wide variety of data integration projects such as tracking sensitive data in the enterprise, application migration, data relationship validation, data set rationalization, and legacy ETL to commercial ETL migration.

The InfoSphere Discovery user interface is Discovery Studio, a task-oriented GUI that presents the steps in order and leads the user through the processes of discovery, analysis, and refining the results.
What InfoSphere Discovery does

IBM InfoSphere Discovery analyzes data values to automate the discovery of relationships between files and tables down to the column level, and automatically discovers the complex transformations required to complete your data integration project. It also finds data inconsistencies across multiple systems and flags errors that can cause significant business issues.

InfoSphere Discovery analyzes data sets, discovering column matches and primary-foreign keys in the source and target data sets. For many data analysis projects, no further discovery is required.

When source-to-target mapping is required, InfoSphere Discovery discovers target matches between the source and target data sets. It then uses these discovered relationships to create maps, which contain the exact SQL statements required to derive each target column from the source data. During this process, InfoSphere Discovery also discovers and reports data inconsistencies within and across the data sets.

InfoSphere Discovery process

The work that InfoSphere Discovery performs in examining the data and detecting relationships is called the discovery process. InfoSphere Discovery automatically discovers this buried information with little or no input from the data analyst. The analyst’s main tasks are to review the discovered results to select the most
appropriate options to use in subsequent actions, and to determine which results would benefit from refinement or re-discovery.

The discovery process can be divided into two phases: analysis and mapping.

**The analysis process**

The analysis process is the sequence of correctly identifying data types and discovering the relationships within each data set (source and target), then using that information to discover the relationships between the source and target data sets.

InfoSphere Discovery performs these steps automatically, but the best results are obtained when you iteratively review the discovered results and approve only the most accurate relationships before proceeding.

**The mapping process**

The mapping process uses the information discovered as a result of analysis to discover the joins, bindings, and transformations that correctly derive the target data from the source data.

Again, InfoSphere Discovery performs these discovery steps automatically, but in order to achieve the highest possible accuracy, you should review and analyze the results, and run additional discovery and refinement steps (such as filters or aggregation) to obtain the most accurate and complete transformations.

---

**Business solutions**

InfoSphere Discovery provides solutions for a wide variety of situations where integrating or understanding data maps, complex relationships, sensitive data flows and transformations between data sets is required.

**Sensitive data flow discovery for security and governance**

InfoSphere Discovery evaluates the data in multiple sources, discovers relationships between tables and files, discovers column level transformations, and documents the relationships as data flows through your organization. You can then take actions to properly secure this sensitive data. This also allows you to verify data movement for internal audits and for compliance reporting (Sarbanes-Oxley, HIPAA, and so on).

**Hand-coded ETL to commercial ETL migration**

InfoSphere Discovery automates the discovery and documentation of the relationships between tables and files in legacy hand-coded integration software and scripts, and automatically generates the source to target data mappings with associated transformations for use by commercial ETL tools.

**Cross-system data analysis**

Before you can implement any data quality, governance, or integration project such as master data management (MDM), data warehousing lineage, or application or data consolidation, you need to know what data you have, where it is located, and how it is related between systems.

For most organizations, the data discovery and analysis process is still manual, requiring months of human involvement to discover cross-system data.
relationships, derive transformation logic, assess data consistency and identify exceptions. The result is a time-consuming and error-prone process that slows time to value, establishes doubt in the accuracy of the data within the new system, and creates the possibility that the new system never succeed in becoming operational.

IBM InfoSphere Discovery is the only data analysis tool that provides a full range of data analysis capabilities including single source profiling, cross-system data overlap analysis, and matching key discovery. It provides an extremely easy-to-use data profiling solution with automatic primary-foreign key discovery and validation, plus the ability to perform cross-source overlap analysis of multiple data sources simultaneously.

With the industry's most advanced features, InfoSphere Discovery is the only profiling solution certified to work with CA ERwin® Data Modeler. It is the ideal profiling tool for data modelers or the part-time data analyst.

Table 1. Cross-system data analysis

<table>
<thead>
<tr>
<th>Cross-source data analysis features</th>
<th>Cross-source Column Overlap Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Performs a cross-compare of all the columns across many data sources (up to 20) in order to establish a baseline of overlapping data across multiple data sources</td>
</tr>
<tr>
<td></td>
<td>• Discovers attribute supersets and subsets</td>
</tr>
<tr>
<td></td>
<td>• Identifies overlapping and unique attributes</td>
</tr>
<tr>
<td></td>
<td><strong>Automatic Primary-Foreign Key Discovery</strong></td>
</tr>
<tr>
<td></td>
<td>• Identifies redundancy and join relationships</td>
</tr>
<tr>
<td></td>
<td>• Discovers and validates primary-foreign key relationships within a single data source</td>
</tr>
<tr>
<td></td>
<td><strong>Profiling/Column Analysis</strong></td>
</tr>
<tr>
<td></td>
<td>Standard column statistics that are discovered automatically:</td>
</tr>
<tr>
<td></td>
<td>• Data Type • Scale • Non-null Selectivity • Mode/Mode% • Data Type Frequency • Formats • Non-null Count • Null Count • Length • Cardinality • Min. • Blank Count • Precision • Selectivity • Max • Value Frequency</td>
</tr>
</tbody>
</table>

IBM InfoSphere Discovery

IBM InfoSphere Discovery provides a complete analysis for prototyping empty targets such as a new data warehouse, MDM hub or data migration. It provides a complete environment for the analysis of multiple data sources and prototyping the combination of those sources into an empty target.

The Unified Schema Builder feature proposes statistically based matching and conflict resolution rules, populates and profiles the resulting target schema before you have to write ETL code for the data warehouse or configure the MDM hub.

InfoSphere Discovery also includes our most advanced transformation discovery capabilities. It automates the discovery of complex transformations and business rules (substrings, concatenations, cross-references, aggregations, case statements,
arithmetic equations etc.) between two structured data sets. It also identifies the specific data anomalies that violate those rules for export to audit and remediation environments, including InfoSphere Information Analyzer Exception Manager.

Table 2. IBM InfoSphere Discovery Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching Key Prototype</td>
<td>Hypothesize and test the quality of matching keys on multiple data sources simultaneously</td>
</tr>
</tbody>
</table>
| Empty Target Modeling and Prototype          | • Drag, drop and combine attributes from data sources to prototype a new unified schema  
                                             | • View the profiling statistics for the prototype target data*                     |
| Precedence Discovery                         | Automatic generation of attribute matching precedence based on statistical analysis |
| Transformation and Business Rule Discovery   | Automated discovery of complex business rules between data sources analyzes thousands to millions of rows of data to automatically discover and statistically validate the following kinds of business rules and transformations between data sources:  
                                             | • Scalar relationships: simple mappings, substrings, constants, concatenations  
                                             | • Inner and outer joins  
                                             | • Aggregations (minimum, maximum, average, sum)  
                                             | • Filters such as equality/inequality, null conditions, conjunctions  
                                             | • Reverse pivot rows  
                                             | • Complex nested case statements (if-then logic)  
                                             | • Cross-reference and lookup tables  
                                             | • Arithmetic relationships between columns |
| Automatic Matching Key Discovery             | Algorithms automatically discover the matching key, including composite key, and statistically validates the key between two data sources |
| Cross-source Data Preview                    | Provides side-by-side preview of data values across multiple data sources for the same logical row. Allows the analyst to see values that match the business rule and anomalies that do not match. |

Methodology

Because of the many ways in which you can use Discovery, there is no single set of instructions that will serve all users.

Many types of data discovery projects can use the tools that Discovery provides to quickly and accurately provide the desired results, and not every project will use Discovery's full capabilities. For example, you can use Discovery to:
- Determine the data types of all data within a data set.
- Discover primary key-foreign key relationships between tables.
- Discover sensitive data wherever it appears in a data set. You can generate a report of that data, and also use Discovery to create a modified data set that masks all sensitive data.
- Discover matching source and target data.
- Discover transformations that, when applied to source columns, accurately transform the data to match the data in target columns.
- Create relevant sample sets for various non-mapping related purposes.

**How IBM InfoSphere Discovery Works**

This section describes the Discovery components in detail, including important concepts and activities. Some of these topics are Discovery Studio actions, and others are general Discovery concepts that you should know before starting a source data discovery (SDD) or transformation discovery (TD) project.

**Discovery components**

Discovery has a client-server architecture containing the following components:

- IBM InfoSphere Discovery server software
- Discovery Studio client software
- Discovery Engine Service software

Each Discovery installation consists of one Discovery Server and one or more Discovery Engine Services and Discovery Studios. These can all be on the same hosts or on different hosts.

Discovery requires a JDBC connection to a [Repository Database](#) and a [Staging Data Source](#) and to the source and target data sources. For more information, see the [IBM InfoSphere Discovery Installation Guide](#).

The following schematic illustrates the Discovery system architecture. The Discovery Engine Services can be deployed in a grid architecture to maximize computing power for discovery.
Discovery Server:

Discovery Server is an intelligent server interface monitoring and managing the interaction among Discovery Studio clients, the Discovery Engine Services and Discovery Engine Processes, a Repository Database and a Staging Data Source. It can simultaneously manage multiple Discovery Studio clients and projects located on different hosts.

Discovery Server provides access to the repository for Discovery Studio to save and retrieve objects. It also stores the discovery results from Discovery Engine processes. When you save or request information, Discovery Server optimizes the associated data access and caching.

Discovery Server runs continuously as a Windows service, regardless of whether Discovery Studio clients or Discovery Engine Services are currently running.

Each Discovery Server requires its own Repository Database. Repositories cannot be shared among Discovery Servers.

Discovery Studio:

Discovery Studio is the client application in which you create, analyze, and manage projects. All of your work in Discovery is performed using Discovery Studio and is saved in the Repository Database.

Each project you create in Discovery Studio requires ODBC and JDBC connections from the Discovery Studio host to a staging data source, and JDBC connections to any Source and Target Data Sources. These connections allow Discovery Studio to read data and to read and write files to the Staging Data Source during processing.
User Interface:

Discovery Studio presents the steps involved in discovery and analysis as a sequence of tabs. Each tab opens a window in which you configure discovery options, analyze data, or verify results related to that step.

In TD projects, Discovery contains the tabs shown below.

In SDD projects, Discovery contains the tabs shown below. Note that the Target Matches and Maps tabs have been replaced with Overlaps, Unified Schema, and Unified Schema PF Keys.

After creating the data sets, you can execute each step individually and review the results one step at a time. When you complete each step, Discovery Studio presents the next step. Alternatively, you can run two or more steps as a single action. You can re-run steps or start over at any time.

Number of Discovery Studios:

A Discovery environment requires at least one Discovery Studio. Several instances of Discovery Studio can be opened on the same host, with a different project open in each. Also, you can install Discovery Studio on multiple hosts in the network with a different project open in each Discovery Studio.

About this task

If there is more than one Discovery environment at a site, you can connect from any networked Discovery Studio to any networked Discovery Server. In order to open a particular project, Start Discovery Studio and specify the Discovery Server that is connected to the repository where the project is saved. A project can be exported from one repository and imported into another.

Discovery Engine Services and Discovery Engine Processes:

The Discovery Engine Service is a scheduling manager that runs as a Windows service. Only one Discovery Engine Service can be installed per host, but a Discovery environment can contain several Discovery Engine Services installed on separate hosts. During Discovery Engine Service installation, you specify a Discovery Server for the service to monitor. This can be changed after installation using the Discovery Engine Configuration Utility.
A Discovery Engine Service creates Discovery Engine processes, also just called Discovery Engines. These are independent processes that execute actions queued by Discovery Studio users. Multiple Discovery Engine processes can be operating at any given time, performing parallel data discovery actions.

How Discovery Engine Processes Work:

When a Discovery Engine is created, it first checks the Discovery Server work queue for actions waiting to be executed. If it finds a task, the Discovery Engine reads the relevant data from the source and target data sources, performs the necessary processing, and then writes the results to the staging data source. At this point, the task is finished.

Then, the Discovery Engine process checks the work queue again. If it finds another task, it executes the task. If it does not find another task, it checks the queue one more time. When a Discovery Engine process has checked the work queue twice in a row and found it empty both times, the process ends.

Frequency and Number of Discovery Engine Processes:

By default, the Discovery Engine Service creates a Discovery Engine process every 30 seconds, 24 hours a day. Executing a task can be very memory intensive, so you can adjust how often each Discovery Engine Service spawns a Discovery Engine process, and also set the maximum number of processes that a Discovery Engine Service will allow at any given time. You can also restrict the Discovery Engine Service to create processes only during certain hours. If you change any configuration settings, you must restart the Discovery Engine Service before the new settings take effect.

For example, you can set the Discovery Engine Service to create a Discovery Engine process every 30 seconds, and to only allow two processes at a time. In this example, if both Discovery Engine processes are executing tasks in the work queue, the Discovery Engine Service will not spawn a third Discovery Engine process until one of the first two processes ends.

Because a large processing job can take several hours, a small task may remain queued for a long time if the maximum number of processes are working on large tasks.

If you have several Discovery Engine Services installed, you set the interval and maximum number of processes on a per-service basis. For example, you may have a Discovery Engine Service installed on the same desktop host as Discovery Server and Discovery Studio, and a second Discovery Engine Service installed on a powerful server. You can configure the Discovery Engine Service on the desktop to create a Discovery Engine process every 30 seconds with a maximum of one process active at a time. The Discovery Engine Service on the server can be configured to create a process every 15 seconds with up to three processes active at a time. In this example, up to four Discovery Engine processes can be running at the same time, and only one of them will be using resources from the desktop.

Using Multiple Discovery Engine Services:

The Discovery Engine Service and Discovery Engines work together to perform all activities queued by Discovery Studio users.
For example, in an environment where several Discovery Engine Services are monitoring the same Discovery Server, actions in the Discovery Server queue may be executed by a Discovery Engine process from any of the Discovery Engine Services. The IBM InfoSphere Discovery Installation Guide discusses ODBC connection naming requirements to handle this.

**Configuring the Discovery Engine Service:**

In a Discovery environment containing several Discovery Engine Services, each Discovery Engine Service is configured individually.

The Discovery Engine Configuration Utility is described in the IBM InfoSphere Discovery Installation Guide. This utility allows you to configure Discovery Engine behavior, including:

- how frequently a Discovery Engine Service checks for existing Discovery Engines.
- the maximum number of Discovery Engines that can exist at a time.
- how long a Discovery Engine waits before checking the queue a second time.
- the hours during which a Discovery Engine Service creates Discovery Engines, which allows you to shift memory-intensive or time-consuming tasks to off-peak hours.

**The Active Window (Scheduling Task Execution):**

The hours during which a Discovery Engine Service creates Discovery Engine processes (allowing it to execute tasks) is called the *active window*. The default start time of 12:00 AM and stop time of 12:00 AM creates an active window of 24 hours a day, meaning that tasks are executed as soon as they are queued.

You cannot schedule a particular task to be executed at a specific time in the future. However, you can change the start and stop time of the Discovery Engine Service using the Discovery Engine Configuration Utility on page 730. Tasks scheduled during the non-active window are held in the queue and executed as soon as the Discovery Engine Service becomes active.

One use for this feature is to allow memory-intensive or time-consuming tasks to be performed outside of normal business hours. This can significantly reduce the time needed to analyze very large data sources. Operation can be maximized by installing several Discovery Engine Services on distributed servers and configuring the active window for off-peak hours.

For example, you can work in Discovery Studio during the non-active window and schedule a number of tasks for execution. When the Discovery Engine Service becomes active later on, Discovery Engines begin executing the queued tasks in order. The results are available in Discovery Studio as soon as each task is finished. If you closed the project and exited Discovery Studio before the active window opened, you can view the results the next time you open the project.

If a Discovery Engine is running at the end of the active window, the Discovery Engine Service terminates it. You can configure the Discovery Engine Service to save the partially completed work. If not, the partial results are lost and the task has to be re-run.

**Checking on Discovery Engine Processes:**
The Discovery Engine Service periodically checks on the status of its Discovery Engine processes. If a process is working on a task or is idle, the service does nothing. If the process has terminated for any reason, the service restarts it and the process checks the Discovery Server task queue. Any interrupted tasks are restarted from the beginning.

Log Files:

Each Discovery Engine Service maintains log files of the tasks executed by its Discovery Engine processes. You can view Status Messages and Log Files to review the actions performed and any messages produced when each task was executed.

Data Sources

A Discovery environment requires connections to at least two database schema:

- A JDBC connection from the Discovery Server to the Repository Database. This is specified during installation, but can be changed later using the Discovery Server Configuration Utility.
- A JDBC connection to each database schema that contains source or target tables, and for any database schema supplying a data rule lookup table.
- A JDBC and an ODBC connection from the Discovery Studio to the Staging Data Source. This is selected when a project is first created in Discovery Studio. If you select a different staging data source after starting to work in the project, you will not be able to access any of the saved results discovered so far.

Note: No connections are required for text files used as data sources or lookup tables.

More information is in the following sections:

- Repository Database
- Staging Data Source
- Source and Target Data Sources
- Data Rule Data Source
- Text Files as Data Sources
- Sharing Physical Databases and Schema

Supported Database Vendors:

A number of database vendors are supported for source, target, and data set data sources. A smaller number of database types are supported for the repository and staging data source.

The list of currently supported database vendors is available from Customer Support.

For details about how Discovery works with specific databases, see Troubleshooting and ODBC Driver Data Type Conversion.

Sharing Physical Databases and Schema:

For performance purposes, the most effective setup is to have all database schema be separate, although they can share the same physical database.

However, in practice, schema can be shared. For example, the repository database and the staging data source can share the same schema.
Repository Database:

Each Discovery Server requires a repository database for the objects created by Discovery Studio users during data processing. Discovery Studio and the Discovery Engine Service communicate with the repository through Discovery Server; they do not directly read from and write to the repository.

The repository database must be the same database type and version as the staging data source. For example, an Oracle repository cannot be used with a DB2 staging data source; an Oracle 10g repository can only be used with an Oracle 10g staging data source.

The machine hosting the Discovery Server needs a JDBC connection to the repository.

The repository database and the staging data source must be one of the Supported Database Vendors.

Staging Data Source:

The staging data source is a database used by Discovery as a work area, where copies of tables and text files in a project are stored and manipulated. This optimizes the processing. The repository database and the staging data source must be one of the Supported Database Vendors.

Discovery creates interim results, called profiling tables, as you process and analyze the data sets. These tables are also stored in the staging data source. Profiling tables are internal and cannot be viewed. Discovery automatically reprofiles tables as needed.

The machine hosting the Discovery Studio that created the project needs an ODBC connection to the staging data source. (The staging data source user is configured in Discovery Studio when creating the project.) To work in that project from a Discovery Studio on another host, that other host must also have an ODBC connection for Discovery Studio’s use, with the same ODBC connection name, to the staging data source.

Each Discovery Engine Service connected to the Discovery Server requires an ODBC connection to the staging data source. All Discovery Engine Service ODBC connections to the staging data source must have the same name. The Discovery Engine Service uses the Discovery Studio’s user logon and password to access the staging data source.

When several projects will use the same staging data source, you can create a Default Staging Data Source so you do not need to retype the same schema information when creating each project.

You can Change the Project's Staging Data Source after a project is created, but you will need to re-run all processing steps.

Source and Target Data Sources:

User data tables and text files to be processed are called data sources. Data sources are added to a project in the Discovery Studio Data Sets window. You can add as many tables or text files as needed for the project.
Text files used as data sources do not require a database connection, but Discovery Studio and the Discovery Engine Service must be able to connect to the hosts where the text files are located.

Discovery has a number of [Supported Database Vendors](#) for source and target tables. A project can contain tables from as many database types as necessary. For example, the source data set can contain Oracle, DB2, and Sybase tables, and the target data set can contain MSSQL and Teradata tables, along with text files.

Each Discovery Engine Service requires a JDBC connection to each source and target database schema.

The repository database and the staging data source must be one of the [Supported Database Vendors](#).

Human review and approval is crucial to determining whether discovered results are valid. At any point in the discovery process, you can override Discovery's results to delete relationships that you know are incorrect, or manually add ones based on your own knowledge of the data. You can also add, delete, and modify expressions, transformations, and binding conditions, and re-run discovery actions with different thresholds to obtain results in a wider or narrower range.

**Data Rule Data Source:**

A data rule table can be imported from any supported source or target database type or version, regardless of the database types used in the source or target, and regardless of the database type used for the repository and the staging data source. The list of supported database types is available from Customer Support.

**Text Files as Data Sources:**

ODBC connections are not required for text files used as data sources. See [About Text Files](#) for more information.

The logged-on Windows user on the Discovery Studio host must be able to access and must have at least read permission for each text file. If the logged-on Windows user does not have read permission, an error will occur during any discovery or processing tasks involving that file.

When Discovery imports a text file, it converts the text file into a table, saves the table in the staging data source, and performs all processing on that table. Almost every processing task is performed on the converted table, and in every Discovery screen the text file displays and behaves as a database table.

Because of this, the term *table* in this document usually refers to both database tables and text files. When it is important to distinguish between a database table and a text file, the difference is clearly noted.

**Password Encryption**

Discovery uses 128-bit encryption for all passwords: the project password in Discovery Studio, all connection passwords within Discovery Studio, and the repository password in the Discovery Server Configuration Utility.
**Processing Data**

This section contains important information about how Discovery processes data.

**Supported SQL:**

Discovery supports standard SQL for most expressions and conditions. Some conditions, such as PF Keys, have certain restrictions; these are clearly described in the documentation where applicable.

**Limitations due to Databases, Drivers, or Third-Party Products:**

Discovery works with data from a number of database vendors and third-party products. Each database and product has unique features and limitations, some of which can affect how Discovery reads, stores, processes, displays, or exports data. For example, a particular database may not support certain SQL constructs or have a limited number of SQL functions available.

Carefully review the documentation for all databases, drivers, third-party products, and other products used as a data source, source of content to be imported into Discovery, or intended recipient of content exported from Discovery. For example, Discovery can work with very long numbers, but a particular driver may not support very long numbers.

If you discover an incompatibility with Discovery, be aware that the results may be affected. You can also modify the data. For example, you may be able to use sample sets or logical tables to filter out incompatible data before starting discovery.

Several known issues that may affect Discovery are described below. Also see "Database Type Considerations" and carefully read all introductory material in each chapter in this book.

**Maximum Field Size:**

The ODBC buffer that Discovery uses is set to 1024 bytes. Fields in database tables or text files that are larger than 1024 bytes will be staged as 1024 bytes.

**Differences in Numeric Precision:**

DB2 allows a maximum numeric precision of 31, but other databases (such as Oracle and Sybase) have a higher numeric precision. When importing columns from other databases that allow larger precision, DB2 may not stage those numbers correctly.

For more information, see Handling long database table numbers when DB2 is staging database.

**Importance of Keeping Data Current:**

It is very important to keep the files in the Discovery data sets current. In an active business environment, changes to database tables can occur frequently: new orders, status updates, changes of address, etc. In order to obtain the most accurate and useful results from Discovery, these changes must be propagated to the imported tables. Reload a Logical Table or Reimport a Physical Table as needed to keep the tables in the staging data source current.
Whenever you queue a task, Discovery checks the staging data source tables to see if they have changed since the last task was executed. This is necessary because the Preview Criteria Screens in Discovery read their data directly from the original source and target files on the network, not from the imported files in the staging data source. Checking the files ensures the data displayed on those screens is always the most current. However, the profiling data that is used in all discovery tasks is read from the imported tables in the staging data source. Discovery automatically re-profiles tables as needed.

If the original files have changed but you have not reloaded or re-imported the staging data source tables, the discovery results may be incorrect.

**Data Profiling:**

When Discovery profiles data, it discovers statistics about each table, such as cardinality, formats, precision, scale, sparseness, and so on. This information is fundamental to the discovery process.

Profiling is performed on the data for the first time during Column Analysis. Tables containing this profile data are stored in the staging data source. From then on, whenever you run or re-run any step, Discovery checks the profile data in the staging data source to see if it is still current. If any change has been made, the affected tables are automatically reprofiled during the execution of the step. This ensures the profile data always remains current.

The Importance of Keeping Data Current cannot be overemphasized. If you do not Reload a Logical Table or Reimport a Physical Table as needed during the Discovery project, the profiling data will not reflect the current content of the tables, and the discovery results may be incomplete or incorrect.

Discovery profiles the data sources in either parallel or serial mode. The difference is in the way the Discovery Server queues the actions.

- In serial profiling (the default), the profiling task for all tables in the source and target data sets is performed as one single action. One Discovery Engine picks up the task and performs all the processing.
- In parallel profiling, the profiling task for each table is performed as an individual item in the Discovery Server queue. Multiple Discovery Engines execute the separate tasks simultaneously, resulting in all tables being profiled more quickly than with serial profiling.

The profiling method is set in the **Profile in parallel** option. The default is false, meaning Discovery profiles in Serial mode.

**The Difference Between Discovered and Defined Results:**

For each step in Discovery Studio, you can have Discovery perform discovery to automatically identify the data types, relationships, expressions, or other information in that step. Alternatively, you can choose to skip the discovery action and instead enter that information manually into the screen. This is useful when you know the data and are confident that running a discovery process in that step would not yield additional useful information. You can also perform discovery and then modify those results by hand, or manually enter more results in addition to the discovered ones.

Information discovered by Discovery is called discovered, and information you enter yourself is called defined. Discovered results that you modify are also considered...
defined. This is an important distinction because in subsequent steps and actions, Discovery treats some discovered data differently than defined data. This provides you with greater control over discovery by allowing you to force Discovery to use your specified relationships, expressions, and so on, which can be helpful in some situations. It also prevents data you have entered or modified from being overwritten if the step is re-run.

For example, say you run discovery for the PF Keys step. After reviewing the results, you modify several discovered PF Keys and then define an additional PF Key. If you now re-run PF Key discovery, Discovery regenerates all discovered, non-modified PF Keys. However, it does not change any discovered PF Keys that you modified, or any PF Keys that you defined.

The ways Discovery treats defined data (compared to discovered data) in each step is described in each step’s Re-Run Step instructions.

About Case Sensitivity:

About this task

Case sensitivity can be turned on or off in each project. This is controlled by the Use Case Sensitive Discovery option when you Create a Project or Edit a Project’s Properties.

- When this option is checked, Discovery is strictly case sensitive. This is the default for Transformation Discovery Projects.
- When this option is unchecked, Discovery normalizes all data to upper case. This is the default for Source Data Discovery Projects.

After performing some or all discovery in the project, you can change the case sensitivity, then re-run all discovery to view the new results.

Note: Some databases, drivers, or other components may not support case sensitivity or case insensitivity. In these situations, you may need to change the default Use Case Sensitive Discovery option so that it is set appropriately.

Case Sensitivity in Searches:

Search dialogs, such as Search Tables or Search Columns, are not case-sensitive, regardless of the project’s Use Case Sensitive Discovery setting.

Data Types Not Supported:

Case insensitivity is not supported in some data types. ODBC Driver Data Type Conversion indicates which data types and databases do not support case insensitivity.

Working With Case Sensitivity:

When the project is case sensitive and you edit any SQL expression or type the name of a schema, table, column, or data rule in Discovery Studio, the name is saved exactly as it is typed, including capitalization. You must understand the case of the tables and columns in your imported data sources, and always be consistent with them as you work in Discovery Studio.

For example, if the table name is HQ_EMP, and the column is LAST_NAME, when entering a database expression in Discovery, you must type

HQ_EMP.LAST_NAME = ‘Smith’
rather than
hq_emp.last_name = 'Smith'

or
HQ_emp.Last_Name = 'Smith'

More examples:
• DBSCHEMA.LastName and DbSchema.LastName are considered two different tables.
• If a column or table name contains a dot or any other special character, the name must be placed in quotes: DBSCHEMA.'Last.Name' = DBSCHEMA.'Last.Name'.
• The expression DBSCHEMA.FirstName=Alex is considered identical to 'DBSCHEMA'.First_Name = 'Alex'.

Internally, Discovery will quote names whenever it is required for SQL generation. For example, when the name contains dot characters, it is quoted.

Reserved Keywords:
By common ODBC standards, there are a number of keywords that are reserved by supported SQL dialects. Avoid using these reserved words in any database-related names. If you must use a reserved word, enclose it in double quotes.

The reserved words are listed in the file <install_directory>\InfoSphere\Discovery\Discovery Engine\Config\ReservedWords.txt.

You can add words to the list, if needed. For more information, contact Customer Support.

Date and Time Formats:
The types of date and time formats discovered are configured during installation. You can change the type after installation, as described in Changing Configuration Parameters.

Fuzzy Matching:
InfoSphere Discovery supports fuzzy matching as a substep in the Overlaps tab and as SQL functions in the Expression Editor.

InfoSphere Discovery supports fuzzy matching from InfoSphere Discovery Studio as a substep of the Overlaps task. You must define at least one of your data sets as a sensitive data repository (SDR) to take advantage of the fuzzy matching step. To refine the results from the fuzzy matching step, you can adjust options for the Overlaps step that relate to the criteria for discovering the fuzzy matches.

Discovery also supports fuzzy string matching entered as SQL functions using the Expression Editor. Fuzzy matching compares strings and allows nonexact matches to be considered matches if the strings meet a minimum threshold of similarity. There are two functions that InfoSphere Discovery provides to do this: DMCOMPARE_EDITDST and DMCOMPARE_LCS.

Fuzzy matching as a substep in Overlaps:
You can propagate approved classifications that have fuzzy relationships from a designated sensitive data repository to columns in other data sets.
About this task

You often need to discover values in data sets with only a general idea of the values that you are looking for. This is especially important in looking for certain values in sensitive data repositories (SDR). The fuzzy matching substep provides you with a way of comparing some subsets of values from a target data set with the contents of an SDR data set.

Procedure

1. From an open project, click the Data Sets tab.
2. Determine which data set that you want to define as an SDR and click the SDR checkbox. To enable fuzzy matching, you must define at least one SDR data set within the project, and only one data set can be designated as an SDR data set.
3. Optional: You can click the Column Analysis tab now so that Discovery can perform some column classification. The classification must be marked as Approved to be considered for other columns.
4. Follow either of these methods to schedule Discovery to use fuzzy matches to find column classifications to propagate:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| From Column Analysis tab | 1. Click the Column Analysis tab and then click Run Next Steps. Make sure that the slider on the Processing Options window is next to the Overlaps step. 
  2. Select the Sub Steps page and select Column Classification: Fuzzy Value Matching |
| From Overlaps tab        | 1. Select the Overlaps tab and then click Re-Run Step.  
  2. In the Column Classification group, click Column Classification: Fuzzy Value Matching |
5. In the Processing Options section, open the New Options window by clicking **New**.

6. In the Generate Overlaps drop-down list, set the values for the following attributes:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum length of the longest common sequence</td>
<td>The minimum number of matching characters in sequence in the SDR column string even if those characters are separated by other characters. For example, if you set this value to 0.8, then at least 80% of the characters in the SDR string must match characters in the string of the non-SDR data set table to be considered a hit.</td>
</tr>
<tr>
<td>Minimum hit rate for fuzzy match</td>
<td>The percent of the total number of rows in the table on the non-SDR side that are hits.</td>
</tr>
<tr>
<td>Minimum length of SDR value for using fuzzy matching</td>
<td>The minimum length of the SDR column value with which to match. This helps determine if the string has sufficient length to warrant the match. A string length that is less than this value will be discovered only as a prefix.</td>
</tr>
</tbody>
</table>
7. Click **OK** and then click **Run**.

**Results**

The classifications that you approved in the SDR data set are propagated to columns in your non-SDR data set by using the fuzzy matching substeps. You can see the results in the Column Classifications view in the **Overlaps** tab.
By using the fuzzy match substep in InfoSphere Discovery, the PN or Personal Name classification is propagated, as shown in the example, from column PA0002.VORNA in the SDR data set to the columns PA0001.SNAME and PA0001.ENAME in the non-SDR data set.

**DMCOMPARE_LCS** (float _min_acceptable_lcs_, String _str1_, String _str2_):

The function **DMCOMPARE_LCS** allows you to compare two strings and considers them a match if they have a minimum number of matching characters in sequence (LCS, or Longest Common Subsequence), even if they are separated by other characters. The minimum number of matching characters is a percentage of the longer of the two strings.

- _min_acceptable_lcs_ is a decimal value between 0.0 and 1.0 that represents the minimum acceptable fraction of the longer string length that must match a string in the shorter string.
- _str1_ is the first string in the comparison.
- _str2_ is the second string in the comparison.

The minimum number of matching characters is determined using the following formula:

\[
_{\text{min\_acceptable\_lcs}} \times \max(\text{length}(\text{str1}), \text{length}(\text{str2}))
\]

If the LCS meets or exceeds the minimum number of matching characters as defined by this formula, the function returns the number of matching characters. If it does not meet the minimum, the function returns 0.

**Examples:**

For example, consider the following expression.

\[
\text{DM\_COMPARE\_LCS}(0.8, '12a45', '12345')
\]

In this example, at least 80% of the characters in the longer string must match characters in subsequence the shorter string to be considered a hit. Both strings have five characters, so the matching number of characters must be at least four (5*0.8 = 4). In this case, four characters do match so the function returns a value of 4 (the number of matching characters).

As another example, consider the following expression:
DK_COMPARE_LCS(0.81, '152b34', '12345')

In this example, at least 81% of the characters in the longer string must be a match with the shorter string. Both strings have five characters so the calculation is $5 \times 0.81 = 4.05$. Four characters match, which does not meet the minimum value of 4.05, so the function returns 0.

For More Information:

This function uses the Longest Common Subsequence (LCS) algorithm, described at:

http://www.iterasi.net/openviewer.aspx?sqlid=73527aa-0c3f-4d0c-82e8-62f72f6b42c

DMCOMPARE_EDITDST (float min_acceptable_distance, String str1, String str2):

The function DMCOMPARE_EDITDST also compares two strings and considers them a match if the number of non-matching characters is not greater than a calculated value.

- $min\_acceptable\_distance$ is a decimal value between 0.0 and 1.0 that represents the maximum acceptable fraction of the longer string length that must be changed in order to achieve a 100% match between the strings.
- $str1$ is the first string in the comparison.
- $str2$ is the second string in the comparison.

The maximum number of characters that must be changed (added, deleted, or modified) is determined using the following formula:

$min\_acceptable\_distance \times \text{max(length}(str1), \text{length}(str2))$

If the number of characters that need to be changed meets or exceeds the maximum as defined by this formula, the function returns 1. If the number is less than the maximum, the function returns 0.

Examples:

For example, consider the following expression.

DMCOMPARE_EDITDST(0.2, '12a45', '12345')

In this example, both strings have 5 characters, so the maximum number of non-matching characters is $5 \times 0.2$, or 1. One character is not a match (a does not match 3), which does not exceed the maximum, so the function returns 1.

As another example, consider the following expression:

DMCOMPARE_EDITDST(0.19, '12a45', '12345')

Both strings again have 5 characters, so the maximum number of non-matching characters is $5 \times 0.19$, or 0.95. One character is not a match (a does not match 3), which exceeds the maximum of 0.95, so the function returns 0.

For More Information:

This function uses the Levenshtein edit distance algorithm, described in the following two places:

http://en.wikipedia.org/wiki/Damerau%E2%80%93Levenshtein_distance
Comparison of DMCOMPARE_LCS to DMCOMPARE_EDITDST:

Both functions provide similar results.

In both functions, performance is non-linear. Processing is faster on short strings. As the string length increases, processing time also increases.

The differences are described in the following table.

<table>
<thead>
<tr>
<th>Difference</th>
<th>DMCOMPARE_LCS</th>
<th>DMCOMPARE_EDITDST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>$O(M \times (N-P))$, where</td>
<td>$O(M \times N)$, where</td>
</tr>
<tr>
<td></td>
<td>$M = \text{length(str1)}$</td>
<td>$M = \text{length(str1)}$</td>
</tr>
<tr>
<td></td>
<td>$N = \text{length(str2)}$</td>
<td>$N = \text{length(str2)}$</td>
</tr>
<tr>
<td></td>
<td>$P = \text{length of the LCS}$</td>
<td></td>
</tr>
<tr>
<td>Threshold</td>
<td>meets or exceeds the minimum number of matching characters</td>
<td>meets or is less than the maximum number of non-matching characters</td>
</tr>
<tr>
<td>Return</td>
<td>$n$ if strings pass criteria</td>
<td>$1$ if strings pass criteria</td>
</tr>
<tr>
<td></td>
<td>$0$ if strings do not</td>
<td>$0$ if strings do not</td>
</tr>
</tbody>
</table>

**Typical usage:**

These functions can be used in any Discovery expression that could yield better results with fuzzy matching. When used in binding conditions or join conditions, the results can be ambiguous.

When used in Unified Schema projects for Source Mapping or as a Matching Condition in Match & Merge Analysis, fuzzy matching can produce large groups because of the transitive nature of fuzzy matching. Processing time can also increase. In these cases, it is recommended to set the Transitive closure option to 0.

When used in Unified Schema projects for Conflict Detection Rules, fuzzy matching can be used to mask conflicts. Performance is good for this usage.

**Primary-Foreign Keys and PF Keys:**

Discovery uses a form of inclusion dependency in the following form:

$Child.PI \narrow Parent.ID$

The right hand side of this inclusion may be a natural key, a generated key, or an explicitly declared primary key. If each value in the right hand side is unique (a selectivity of 1), the inclusion is a potential foreign key. While the relationship may be incidental, it looks exactly like a primary-foreign key relationship.

Since the most well-understood example of such inclusion dependency is the primary-foreign key relationship, this is referred to as a PF key in Discovery. The right hand side is the primary side, and the left hand side is the foreign side.

For more information, see Discover PF Keys.
Overview

IBM InfoSphere Discovery allows you to create two types of projects:

- Projects with a goal of discovering overlaps or creating a unified schema are created as Source Data Discovery Projects, also called SDD projects.
- Projects with a goal of transformation discovery are created as Transformation Discovery Projects, also called TD projects.

The first steps in each type of project are the same. You create and populate data sets, discover data types, discover primary-foreign keys, and organize the data into data objects within each data source. After this point, different discovery and analysis options are available.

Source Data Discovery Projects

In SDD projects, you review data value overlaps between tables within and across data sets. In addition, you can create a unified schema and discover unified schema primary-foreign keys.

Results include a summary of the total number of tables and columns in the data sets, number of exclusive columns, percent of value overlap, number of tables and columns containing overlapping data, and more detailed statistics, including views of the actual overlapping data itself.

Transformation Discovery Projects

In TD projects, you discover transformations between a source data set and a target data set. The Types of Relationships Discovered include joins, transformations, binding conditions, and data rules.

A related product, IBM InfoSphere Information Analyzer Exception Manager, provides ongoing validation of transformations, bindings, and PF keys between specified source and target data sets, and allows review and remediation activities.

Common Activities in All Projects

The activities described in this section are single-source analysis actions common to all projects. These steps establish good quality metadata that will be used to guide cross-source analysis. Later sections in this chapter describe the subsequent steps for SDD and TD projects separately.
### Scope Data, Normalize Data Types, and Perform Column Analysis

<table>
<thead>
<tr>
<th>Corresponding Discovery Studio Step</th>
<th>Create a Project</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Create Data Sets</td>
</tr>
<tr>
<td></td>
<td>About Sample Sets</td>
</tr>
<tr>
<td></td>
<td>Perform Column Analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Objectives of this step</th>
<th>Create data sets that are relevant and manageable in size, with accurate data types</th>
</tr>
</thead>
</table>

| Why should you do this? | • A large schema may contain a lot of irrelevant data.  
                          | • A large data set will reduce performance  
                          | • Inaccurate data or data type specification reduces the accuracy of various discoveries. Correcting this now makes subsequent discovery cleaner and easier. |
|-------------------------|-----------------------------------------------------------------------|

| What are the goals in this step? | • Identify the relevant tables within a large schema  
                                  | • Verify all data types are correct  
                                  | • Discover column statistics  
                                  | • Create sample sets |
|-----------------------------------|-----------------------------------------------------------------------|

| What does Discovery do? | • Import tables into the source and target data sets  
                          | • Discover date and numeric data types that appear as string types, for both database tables and text files  
                          | • Display data statistics |
|-------------------------|-----------------------------------------------------------------------|

| What do you do? | • Search tables and text files in schema by table and column name  
                   | • Import tables and text files into the source and target data sources  
                   | • Preview data in the imported tables and text files  
                   | • Execute data type discovery  
                   | • Modify discovered data types as necessary  
                   | • Review data statistics  
                   | • Create sample sets |
|-------------------|-----------------------------------------------------------------------|

In the Column Analysis step, data type discovery is performed for both database tables and text files. It is essential for files because these do not contain metadata. Tables can contain metadata, but also may have inaccuracies.

There are two aspects of data type discovery that need to be considered before performing column analysis:

• How much data needs to be scanned? If the database table contains 10 million rows, do we need to scan all the rows to accurately determine the data types? If not enough rows are scanned, the discovered data type may not be accurate.

• How tolerant is the typing? For example, if a column contains mostly numeric values and a very small number of alphanumeric, should we consider its type to be numeric or string?

You can adjust Discovery’s discovery thresholds to accommodate the particular requirements of your data. If the correct data types are not discovered, not all data values in the column will conform to the type. These exception values cause formatting errors later in the discovery process.
It is important that you verify each discovered data type to determine if accurately describes the values in the column. You may need to manually modify some of the data types.

**Discover and Analyze Primary-foreign Keys**

<table>
<thead>
<tr>
<th>Corresponding Discovery Studio Step</th>
<th>[Discover PF Keys]</th>
</tr>
</thead>
</table>
| **Objectives of this step**        | • Make implicit relationships explicit  
                                   | • Logically cluster tables into “data objects” |
| **Why should you do this?**        | • PF keys encode relationships among tables. Making these relationships known to Discovery allows Discovery to perform additional types of discovery, such as join conditions.  
                                   | • Tables and maps organized around data objects are more meaningful, easier to review, and provide more knowledge to Discovery, allowing for more accurate discoveries. |
| **What are the goals in this step?**| • Establish meaningful PF key relationships  
                                   | • Establish data objects that are semantically correct |
| **What does Discovery do?**        | • Discover PF Keys  
                                   | • Display PF key statistics for discovered and hand-entered PF keys  
                                   | • Discover data objects  
                                   | • Create data object diagrams and display statistics for reviewing and modifying data objects |
| **What do you do?**                | • Execute PF key discovery and manually enter any known PF keys  
                                   | • Review PF keys and statistics  
                                   | • Execute data object discovery  
                                   | • Review and modify data objects |

Metadata does not encode complete semantics, either because the metadata data model used does not have the expressive power to encode such semantics, or because the semantics are not explicitly declared. For example, data analysts frequently do not declare primary or foreign keys to avoid a performance penalty; they prefer to control the integrity of the data using application logic.

Metadata on its own can be interpreted in many different ways and is seldom accompanied by thorough documentation. There may be several experts in the organization or existing documentation, but usually the experts are working on newer projects and have forgotten the details, or the documentation is difficult to locate or has not been maintained. In most cases, metadata simply is not sufficient.

As with any mapping process, the more Discovery knows about the semantics, the more accurate the final results will be. Discovery communicates with you using terms and concepts that are intuitive and familiar, and through easy steps that gradually solicit implicit semantics.

We identify two kinds of information that are often missing but which are major carriers of semantics, and are hence critical in building correct mappings:

• PF keys
• Data objects
Discovery automatically discovers PF keys, including composite PF keys, and data objects. You review, analyze, and verify the discovered items. Discovery provides visualization and a set of intuitive statistics to help you validate these discovered artifacts. By analyzing PF keys and data objects to the best of your abilities, you provide critical semantics that guide the later discovery of join conditions, and also enable more focused analysis such as creating mappings between source and target data objects that semantically “match”.

While PF keys and data objects by themselves are a useful way to understand a data set, your ultimate goal is usually creating valid maps. For that goal, Discovery performs PF key and data object analysis to automatically execute table matching and join condition discovery. This is critical if you have no other way of forming these matches and defining the join conditions.

However, if you already know these relationships, you do not need to discover PF Keys or data objects, and can just perform cross-source profiling, manually create maps, manually specify join conditions, and then use the transformation profiling and analysis facilities to build mappings. This way, you do more work using your knowledge instead of leveraging the discovery capabilities of Discovery.

The different types of PF Keys are described below.

**Partial PF Keys**
A perfect PF Key ($fp$) has the following statistical properties:

- $p$ has a selectivity of 1
- $f$ has a hit rate of 100%

A partial PF Key fails on one or both of these conditions. Partial PF Keys can be caused by dirty data but can still be meaningful. Whether or not to consider partial keys as keys, and the degree of deviation to allow, depends on the actual mapping context: if you know the data is absolutely clean, then do not allow partial keys. You can provide this information to Discovery by modifying PF key thresholds.

**Incidental PF Keys**
Even if a PF Key has perfect statistics, it may not be a meaningful PF Key. Such PF Keys are referred to as incidental PF Keys. For example, if column A contains unique sequential numbers from 1-10000 and column B contains a numeric code of 1-4 digits, Discovery discovers $B\|A$ as a PF Key, but in reality, the match is incidental.

Incidental PF keys often have a very low hit rate on the foreign side, and you can modify the Discovery PF Key thresholds to keep Discovery from discovering these. Even with those changes, though, some of the PF Keys that Discovery discovers are likely to be incidental and will need to be removed.

Discovery provides powerful features to help you identify and remove incidental PF Keys:

- In the PF Keys grid, you can quickly identify and remove all PF Keys with a particular column as the primary or the foreign key, if you are certain that column is not a meaningful key.
- You can remove all PF Keys with a foreign side hit rate below a certain value.
- By filtering the results and then sorting the data by hit rate, you can further identify incidental keys.
Organize each Data Set into Data Objects

Corresponding Discovery Studio step: Discover Data Objects.

If you know which source and target tables map to each other, you do not need to generate data objects but can define maps manually. If you do not know how the source and target tables relate to each other, use Discovery to discover data objects.

Transformation Discovery (TD Projects)

After completing the Common Steps for All Projects, you are ready to perform the TD-specific steps.

The goal is a set of SQL transformations that transform source table column values into target table column values. The IBM InfoSphere Discovery Quick Reference is a flowchart that assists you in determining which Discovery steps need to be performed based on the project goal.

Discovery discovers mappings by comparing data in a source to data in a target. The source and target must contain data about the same entities.

For example, a set of target data may have been derived from a set of source data. If the transformations were poorly documented or lost, it is extremely difficult to reconstruct by hand the SQL expressions that transformed the source data into the target data. Discovery discovers these lost transformations by comparing the source ‘before’ data to the target ‘after’ data.

As another example, Discovery can discover the relationships between two customer masters containing the same customer population. The two data masters may use different schema, metric systems, terminology, and so on, but the data in each one describes an overlapping, if not identical, population.

In general, Discovery finds the relationships between a source and a target in the following scenarios:

- **The source and target contain exactly the same data**, but it is modeled differently. This is the scenario in which Discovery yields the most accurate discovery results.

- **The source and target data overlap**, but are not exactly the same. Discovery contains tools for creating and managing sample sets, so by creating source or target sample sets containing exactly the same data, you can perform discovery with results as accurate as in the previous scenario.

  If you cannot obtain acceptable sample sets, some of Discovery’s analysis results are less accurate. However, the statistics that Discovery computes for the data can still help you, if you interpret the results by factoring in known qualities of the sample set.

- **The target has no data** (the empty target scenario). Discovery can analyze and compute some statistics for the source data set, but its source-to-target data mapping abilities are not used in this scenario. You can still obtain results by manually defining a small amount of target data for Discovery to use for mapping. You can also define transformations and generate queries and ETL scripts.

Mapping with Discovery is a semi-automatic process where Discovery progressively discovers deeper properties of the data that eventually lead to the
final result: a detailed mapping with transformations. Mapping is performed as a sequential, multi-step process. The general progression is:
1. Discover properties.
2. Review, verify, and modify the discovered properties.
3. Discover next properties (or rediscover the current properties after modifications).

To help you manage the complexity of the exercise, we designed Discovery with the following goals in mind:
- Guide you through a set of clear and progressive discovery steps, each one building on the last step.
- In each step, allow you to manipulate the discovery results in such a way that you can quickly focus on critical pieces of information.

Discovery follows an underlying philosophy, described in the next sections, to determine the sequence of events and the actions performed on each screen. Understanding this philosophy will allow you to better understand the activities performed with Discovery and to use its capabilities most effectively.

The activities performed in Discovery are:
- **Discover Source-to-target Value Matches**
- **Discover Source-to-target Matches**
- **Discover and Refine Source-to-target Column Transformations**

**Discover Source-to-target Value Matches**

<table>
<thead>
<tr>
<th>Corresponding Discovery Studio Step</th>
<th>Generate Target Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objectives of this step</td>
<td>Baseline measurements of cross-source value overlap</td>
</tr>
<tr>
<td>Why should you do this?</td>
<td>First time to see how source and target data match up</td>
</tr>
<tr>
<td></td>
<td>Shows cross-source relationships as value overlaps.</td>
</tr>
<tr>
<td></td>
<td>Required if you need to have Discovery discover binding conditions; can skip otherwise</td>
</tr>
<tr>
<td>What are the goals in this step?</td>
<td>Calculate cross-source value matches (target matches)</td>
</tr>
<tr>
<td>What does Discovery do?</td>
<td>Discovers target matches</td>
</tr>
<tr>
<td>What do you do?</td>
<td>Review target matches</td>
</tr>
<tr>
<td></td>
<td>Delete as many meaningless target matches as possible; remaining meaningless target matches may affect outcome of map and binding condition discovery.</td>
</tr>
</tbody>
</table>

So far we have been working with individual data sets, gathering pieces of semantics and recording them in the form of valid PF keys and data objects. Cross-source analysis is more difficult than single-source analysis, because we need to deal with the differences, similarities and relationships between two sets of semantics. Even though you have worked to improve the accuracy of each set of semantics, matching the semantics across data sets introduces another level of complexity.

Discovery makes only one source-to-target measurement: target matches, containing all of the source and target column pairs that overlap in value. We compute
various properties of the overlap and compile a large list of source-to-target column matches. Target matches of several categories are found with these rules, including:

- Source column $S$ and target column $T$ overlap in value because they are “the same”; for example, they both contain social security numbers of the same customers.
- Source column $S$ and target column $T$ overlap in value because they draw values from the same domain; for example, they both contain US addresses. However, $S$ may be the supplier address, while $T$ may be the client address.
- Source column $S$ and target column $T$ overlap in value but are actually unrelated to each other; for example, $S$ contains code values in the range of 1-150, and $T$ contains sequential numbers in the range of 1-10000.

Target matches are simplistic measurements that indicate, not promise, valid cross-source semantics. For example, if a source column $Source.Customer.AcctNumber$ is in every way “the same” as a target column $Target.CU.AcctID$, the target match statistics indicate a 100%, 1-to-1 match between the columns. However, even if two columns overlap in value significantly, they may have nothing in common semantically.

Depending on your goal, you can use target matches in different ways:

- To identify sensitive data. For example, suppose you are looking for all appearances of social security numbers in a particular format. If at least one column in the data set contains social security numbers in that format, target matching locates all other columns that contain matching values in that format. In this case, all discovered target matches are useful.
- To achieve higher accuracy and improve performance. In later discovery steps, Discovery analyzes target matches to determine which ones are valid and which are merely incidental. By identifying strong relationships and removing weaker ones now, you reduce the number of possibilities for Discovery to explore during subsequent processing.

Typically, we recommend that you review the target matches as much as possible, and remove obviously meaningless matches. However, if you do not have enough knowledge to perform this review, you can leave questionable matches in place. Discovery will ignore the weaker ones during discovery. As in all situations, the more you can do now, the more accurately Discovery will perform.

### Discover Source-to-target Matches

**Table 3. Discover source-to-target matches**

| Corresponding Discovery Studio Step | • **Discover Maps**  
| | • **Discover Joins**  
| | • **Discover Binding Conditions**  
| | • **Discover Where Clause**  
| Objectives of this step | • Baseline measurement of cross source value matches (target matches )

Chapter 2. Getting started with InfoSphere Discovery projects 31
Table 3. Discover source-to-target matches  (continued)

| Why should you do this?                                      | • This is the first time to see how source and target data match up  
|                                                            | • Great way to understand cross-source relationships as value overlaps  
|                                                            | • If you plan to let DM discover binding conditions, you must run this step  
|                                                            | • If you already know binding conditions, skip this step |

| What are the goals in this step?                           | • Calculate target matches |

| What does Discovery do?                                    | • Compute target matches |

| What do you do?                                            | • Review target matches and clean out meaningless matches as much as possible  
|                                                            | • Remaining meaningless target matches may affect the outcome of map and binding condition discovery |

---

Discover and Refine Source-to-target Column Transformations

Mapping and analysis is the core strength of Discovery. Discovery offers a rich set of capabilities that help you discover and calibrate transformations that derive a target table from one or more source tables. The basic paradigm for mapping in Discovery is transformation discovery driven by binding conditions and correlations.

| Corresponding Discovery Studio Step | • Discover Transformations  
|                                   | • Generate Data Rules  
|                                   | • Discover Reverse Pins |

| Objectives of this step            | Build table level maps |

| Why should you do this?            | This step produces the actionable mappings. |

| What are the goals in this step?   | Establish a way of deriving the target table from the source tables. |

| What does Discovery do?            | • Data  
|                                   | • Discover join conditions  
|                                   | • Discover binding conditions  
|                                   | • Discover transformations  
|                                   | • Discover data rules (lookup tables) |

| What do you do?                    | • Execute join condition discovery  
|                                   | • Review join conditions using statistics and hit/miss data views  
|                                   | • Execute binding condition discovery and binding condition filter discovery  
|                                   | • Fine-tune binding conditions using statistics and hit/miss data views  
|                                   | • Execute correlation discovery  
|                                   | • Execute transformation discovery, including a rich variety of related discoveries  
|                                   | • Execute transformation filter discovery  
|                                   | • Fine tune the transformation for each target column using statistics and hit/miss data views  
|                                   | • Run additional discovery tasks to refine transformations  
|                                   | • Execute aggregation and reverse pivot discovery |
A map describes how data from source tables is transformed into data in a target table. Mapping in Discovery is much more than simply generating a SQL query; it is a complex concept involving join conditions, filters, and column transformations.

Binding conditions and cross-source correlations are a concept unique to Discovery. When we ‘bind’ a source table and a target table, we are matching source records to target records using a binding condition, usually an equijoin on natural keys. After binding, we compare the matching source and target records in order to discover transformations for the columns that were not used in the binding condition. This is best illustrated by a series of examples, shown below.

As shown below, the mapping semantics can be displayed visually. Various parts of the map, such as join conditions, column transformations, and where clauses, can be discovered and established using a variety of tools, including extensive statistics analysis, data previews, and manual prototyping.
Building maps is not a new concept. Many mapping products in the industry construct maps visually, although few provide data-driven discovery and verification of such maps. One of the many unique features provided by Discovery alone is the use of extensive statistics and data preview to verify and fine-tune the results.

The following table compares Discovery’s mapping abilities with more traditional approaches.

<table>
<thead>
<tr>
<th>Source-to-target ID</th>
<th>Discovery</th>
<th>Other Mappers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data object-driven, data driven</td>
<td>Manual</td>
</tr>
<tr>
<td></td>
<td>Discovery</td>
<td>Other Mappers</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Join condition discovery</td>
<td>Based on explicit or implicit PF Keys; can also use name matching and data preview</td>
<td>Based on explicitly declared foreign keys or using name matching</td>
</tr>
<tr>
<td>Join condition analysis</td>
<td>Extensive statistics to characterize the join, including hit rate, overlap, and data preview</td>
<td>No</td>
</tr>
<tr>
<td>Column correspondence discovery</td>
<td>Binding condition driven correlation, hit/miss data preview</td>
<td>Name matching</td>
</tr>
<tr>
<td>Column transformation discovery</td>
<td>Automatic discovery of a wide variety of common transformations</td>
<td>No</td>
</tr>
<tr>
<td>Column transformation analysis</td>
<td>Hit and miss analysis, data preview</td>
<td>No</td>
</tr>
<tr>
<td>Domain translation tables (Data rules)</td>
<td>Automatic discovery and analysis with data preview. Data rules used for various transformations.</td>
<td>No</td>
</tr>
</tbody>
</table>

**Creating Maps**

A map identifying a target table and one or more source tables can be created by hand if you know the data well enough.

When schemas are large, poorly documented, or unfamiliar, you may not know the relationships between the source and target tables very well. In these cases, Discovery will automatically discover maps based on data objects created from the source and target data sets. After the map exists, Discovery performs extensive analysis to discover the binding condition, transformations, where clauses, filters, and other relationships.

Maps are created in two ways:

- **Manually.** To do this, you have to know exactly which source tables map to which target tables. If multiple source tables need to be joined together to map to a target table, you can manually create join conditions and then having Discovery calculate statistics for them. Or, if you ran PF Key discovery for the source data set, Discovery will discover the join condition.
- **Automatically.** Discovery will create maps (possibly many) and discover join conditions if you ran column analysis and discovered PF Keys, data objects, and target matches.

The advantage of working with data object-driven maps is using the deep technology to work with data fragmentation caused by normalization. Sometimes this fragmentation is logical and sometimes it isn’t. When it is not, it can be quite tricky to figure out the original relationships. Instead of trying to determine the table relationships manually, you can work with data objects and target maps to gradually discover the deep semantics needed to create meaningful maps.

**Working with Join Conditions Using Statistics and Data Views**

If a map has multiple tables on the source side, a join condition that correctly joins the source tables is required. If PF Keys exist, either from discovery or from manual definition, Discovery will use them as building blocks to discover join conditions. You can also define join conditions by combining existing PF keys, or simply by entering a SQL expression. Regardless of whether the join condition was discovered or defined, Discovery will compute statistics on the join, as well as allowing you to preview the data to understand the meaning of the join.
Discovery provides the following facilities to help you create joins:

- All PF keys between source tables are presented as a list of join matches. You can use these as building blocks for creating join conditions. Discovery provides drag and drop and context sensitive editing. Complex join conditions involving transformations can only be typed in by hand.
- Once a join condition exists, either typed in by hand, composed using join matches, or discovered by Discovery, you can generate statistics and hit/miss data previews for the join condition. The statistics and the data previews help you understand the properties of the join condition, such as how many rows in each relation are preserved or lost by the join. You also need to look at the data after the join is applied, to understand whether the join is meaningful.

### Binding Conditions

Binding conditions are relationships between a target table and one or more source tables, based on a natural key or data rule that relates or binds the data in one or more source column to the data in one or more target columns. Binding conditions do not exist independently, but only in the context of a map, given all the properties of that map.

Binding conditions can be simple or complex. For example, a binding condition may be a simple equality relationship between a column in a source table and one in a target table. Or, it may be based on a transformation involving several source columns and one target column.

The distinguishing features of Discovery are automatically discovering binding conditions, and computing correlations based on these binding conditions. In general, the search for potential binding conditions is guided by a rich set of heuristics, which also assist in selecting the binding condition that has the most potential to be semantically correct. Discovery also considers data rules when discovering binding conditions.

Correlations are based on binding conditions. If you change a binding condition, you need to recompute correlations. Correlations are an important measurement that are the basis for all transformation discovery. A good binding condition creates a high number of correlations. For the most part, binding conditions tend to involve natural keys, which often have good selectivity and high cardinality. Also, a better binding condition binds more rows than a poorer one.

Discovery discovers three kinds of simple (equality) binding conditions between the source and target: one to one, one to many, and many to one.

- If the binding condition is one to one, each record in the source is bound to one record in the target.
- If the binding condition is one to many, each record in the source is mapped to several records in the target. In this case, you can have Discovery attempt to establish a reverse pivot transformation.
- If the binding condition is many to one, where multiple records in the source map to one record in the target, you can have Discovery attempt to establish an aggregation transformation.

Once a potential binding condition is identified, Discovery presents a set of statistics that illustrate the properties of the binding condition, including hit rates on both the source and target side, source/target rows that are bound or unbound by the condition, and cardinality and selectivity of the binding columns.
You can modify the binding condition to improve the statistics. If modifying the binding condition does not sufficiently improve the results, create a new one. A good way to start is by browsing the list of target matches for one-to-one matches with a high row hit rate and selectivity.

After modifying or creating a binding condition, rediscover correlations and all other column level relationships related to that binding condition.

**Binding Condition Guided Analysis**

The following expression is an example of a discovered binding condition.

\[ S.fn = T.first\_name \text{ and } S.ln = T.last\_name \]

As with all discoveries, a binding condition can be statistically perfect but not semantically meaningful. One cause of this is an irregularity in the data used for the analysis. After examining a discovered binding condition, you may decide it is not meaningful or not valid for some other reason. In this case, you need to manually define a binding condition in order to proceed with automatic discovery.

Discovery provides several tools to help you explore, test, and validate binding conditions. The following examples illustrate these tools. Note there is no science in these exercises; they are explorations using Discovery tools and common sense, along with any context information you have.

**Example 1**

This is the simplest example of binding conditions and column transformations. This example contains 2 tables.

| Table 4. Example 1: binding conditions and column transformations |
|---------------------------------|-----------------|
| **Source** | **Target** |
| **Id** | **Name** | **Dept** | **Territory** | **Id** | **Name** | **Dept** | **Territory** |
| 1 | Bill Jones | 12 | Atlantic | 1 | Bill Jones | 12 | Atlantic |
| 2 | Mary May | 22 | WestCoast | 2 | Mary May | 22 | WestCoast |
| 3 | Laura Li | 12 | Atlantic | 3 | Laura Li | 12 | Atlantic |
| 4 | Bob Box | 17 | Atlantic | 4 | Bob Box | 17 | Atlantic |
| 5 | Fred Foe | 11 | Atlantic | 5 | Fred Foe | 11 | Atlantic |

The binding condition for these tables is \( \text{Source.Id} = \text{Target.Id} \). By binding on the \text{Id} column, we discover the following three equality column transformations:

- \( \text{Target.Name} = \text{Source.Name} \)
- \( \text{Target.Dept} = \text{Source.Dept} \)
- \( \text{Target.Territory} = \text{Source.Territory} \)

**Example 2**

This example uses the same binding condition as the previous example, \( \text{Source.Id} = \text{Target.Id} \).

| Table 5. Example 2: binding condition is Source.Id=Target.Id |
|---------------------------------|-----------------|
| **Source** | **Target** |
| **Id** | **Name** | **Dept** | **Region** | **Id** | **Name** | **Dept** | **Territory** |
| 1 | Bill Jones | 12 | Eastern | 1 | Bill Jones | 12 | Atlantic |
| 2 | Mary May | 22 | Western | 2 | Mary May | 22 | WestCoast |
Here, the source column Region corresponds to the target column Territory, but they contain different values. Instead of a simple equality transformation, Discovery first discovers the correlation Source.Region ↔ Target.Territory.

This correlation is made into the following data rule, which is applied to the Region column in order to derive the Territory column values:

**Table 6. Correlation of Region and Territory**

<table>
<thead>
<tr>
<th>Source.Region</th>
<th>Target.Territory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eastern</td>
<td>Atlantic</td>
</tr>
<tr>
<td>Western</td>
<td>WestCoast</td>
</tr>
<tr>
<td>Central</td>
<td>Atlantic</td>
</tr>
</tbody>
</table>

### Example 3

Binding conditions are not always as obvious as in the previous two examples. In the example below, Source.Id=Target.Id appears at first to be a good binding condition. But it yields no correlations, so it is not a valid column for a binding condition.

**Table 7. Example 3: Binding condition using a column that is not valid**

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Name</td>
</tr>
<tr>
<td>1</td>
<td>Bill Jones</td>
</tr>
<tr>
<td>2</td>
<td>Mary May</td>
</tr>
<tr>
<td>3</td>
<td>Laura Li</td>
</tr>
<tr>
<td>4</td>
<td>Bob Box</td>
</tr>
<tr>
<td>5</td>
<td>Fred Foe</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Name</td>
</tr>
<tr>
<td>1</td>
<td>Fred Foe</td>
</tr>
<tr>
<td>2</td>
<td>Bill Jones</td>
</tr>
<tr>
<td>3</td>
<td>Bob Box</td>
</tr>
<tr>
<td>4</td>
<td>Mary May</td>
</tr>
<tr>
<td>5</td>
<td>Laura Li</td>
</tr>
</tbody>
</table>

The correct binding condition is Source.Name=Target.Name, which Discovery automatically discovers. These columns yield the following correlations:

- Source.Region=Target.Territory
- Source.Dept=Target.Dept
- Source.Id=Target.Id

### Example 4

Binding conditions are not always 1-to-1, where one source row binds to exactly one target row. Multiple source rows may bind to a single target row. When this happens, we say the relationship cardinality of the binding condition is many-to-1.

In such cases, Discovery attempts to discover aggregation functions as column transformations.
Table 8. Example 4: Binding condition is many-to-1 (continued)

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Bonus</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Bill</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Bill</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Mary</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>Mary</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>Mike</td>
<td>3</td>
</tr>
</tbody>
</table>

In this example, the binding condition is `Source.Id=Target.Id` and the cardinality is many-to-1. The following transformations are discovered:

- `Target.Name=Source.Name`
- `Target.Bonus=sum(Source.Bonus)`

**Example 5**
This example describes refining existing binding conditions, using data previews. You have a source table, `S`, and a target table, `T`. Discovery finds a binding condition of `BC S.company_name = T.name`, with a target hit rate of 98%. Using the binding condition Show Misses data preview, you see that 2% of the target rows are not bound. Some of the missed rows are source, and some are target.

To understand why these rows are not bound, in the Show Misses view, sort the target table by the `name` column, and the source table by `company_name`. This reveals that the unbound rows contain very long company names, and shows that the source and target columns have different lengths: `T.name` allows 30 characters, while `S.company_name` allows 40 characters. Discovery considers a 30-character string and a 40-character string as misses.

Based on this information, you modify the binding condition to the following and then refresh the statistics. This yields a target hit rate of 100%.

```
substr(S.company_name, 30) = T.name
```

**Example 6**
This example describes searching for new binding condition using target matches and data previews. Assume you cannot improve a discovered binding condition, so you look at the target matches for suggestions. The target matches `S.fn` and `T.first_name` have good statistics. You enter the following as a binding condition and then refresh statistics:

```
S.fn = T.first_name
```

The resulting statistics indicate this is a many-to-many binding condition. You are looking for a one to one binding condition and believe you can add another condition to this one to make it one to one.

To find a column to add, you display the Show Data data preview. When you focus on the `S.fn` value `Paul`, you see that it is bound to three `T.first_name` rows, all with the value `Paul`.

You now drag more source and target columns into the Show Data view and focus on `S.fn = Paul` and `T.first_name = Paul`. Looking at the additional columns, you find that `S.in` and `T.last_name` have unique values among the source and target rows. You add this information to the binding condition as shown:
(S.fn = T.first_name) AND (S.ln = T.last_name)

After refreshing statistics, Discovery identifies this as a one-to-one binding condition with a 100% hit rate.

**Correlations**

Discovery can discover multi-column correlations. For example, assume two columns, C1 and C2, and two rows, r1 and r2. They are related as shown:

if r1[C1] = r2[C1] then r1[C2] = r2[C2]

Basically, specific values of C1 always appear on the same row as specific values of C2. We can infer that C2 is correlated to C1 and that C2 is likely computed from C1 using a transformation. In multi-column correlation, C1 is considered as not just one column, but a composite column that includes both itself and C2.

**Working with Column Transformations**

Once a binding condition is established, Discovery compares the source data to the target data for each column in the target data source. This produces a variety of transformations for each target column, along with statistics for each transformation. Generally, a correct transformation is at or near to a 100% hit rate. You can raise the Transformation Discovery thresholds to restrict Discovery to discovering only transformations with high hit rates.

Transformations that are statistically perfect may still be incidental or semantically meaningless. To further confirm the semantics of a discovered transformation, use the Show Hits/Misses data previews. This reveals the finer details of a transformation using the actual data.

You can also manually enter a transformation and have Discovery calculate statistics and Show Hits/Show Misses data previews for it. This way, if none of the discovered transformations are good enough, you can experiment with various expressions and use Discovery to assess the correctness of each one.

Based on a good binding condition, Discovery computes correlations between all source columns and target columns. Then, we can do several things:

- Generate data rules.
- Use known data rules to translate the source column data to the target column data.
- Discover a variety of common column transformations:
  - Arithmetic functions
  - String operations
  - Common aggregation functions
  - Conditional transformations (case statements)

The basis for verifying transformations is the hit rate statistics and the hit/miss data previews. Hit rate statistics are provided for each column transformation and its related binding conditions. Hit/miss data previews are computed for all types of transformations, including aggregation functions and data rules.

If a discovered transformation is correct, its statistics indicate high hit rates. Good transformations show good statistics, but the reverse is not true; good statistics do not always indicate good transformations.
Discovery discovers all transformations with good statistics and presents them as results. You need to review and verify all discovered transformations to confirm the semantic correctness of each one.

In the following example, the binding condition is \texttt{Source.Key=Target.Key}. The discovered transformation is \texttt{Target.X=Source.A+Source.B}. The hit rate of 77\% shows that seven out of the nine target columns are correctly derived with this transformation. By using Show Misses, you can identify the exact rows in the target table, shown in bold, that do not match the transformation.

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>A</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td>40</td>
</tr>
</tbody>
</table>

**Complex Transformations**

Discovery discovers complex transformations, such as case statement transformations and data rule discoveries.

Typically, a case statement transformation has a substantial but unsatisfactory hit rate, such as 30\% to 70\%. This indicates a significant portion of the source columns are not being mapped, and this may be a conditional transformation. To improve case statements, manually run filter discovery to raise the hit rate until it reaches 100\%, or until it stops improving.

An example of a case statement is shown below.

\[
\text{CASE WHEN HQ\_EMP\_TERMINATION\_DATE is null THEN 'C' WHEN HQ\_EMP\_STATUS = 'Resigned' THEN 'R' ELSE 'F' END}
\]

**Data Rules**

Data rules are a special kind of transformation based on a lookup table. They can be discovered automatically when Discovery detects transformations that are too complex to be represented by a single SQL statement, or whose SQL statement would be unmanageable. You can edit and refine data rules once they are created, or you can create them yourself manually.

You may be able to map an unmapped target column using a data rule. Data rules are generated manually from correlations. When a target column has a reasonably low cardinality and a very strong correlation with a particular source column, you can generate a data rule from this correlation and use it as a transformation to map the columns.
Discovery often discovers many strong correlations. You need to use your intuition and any background knowledge to decide whether the columns in a correlation are indeed the same information presented in different encoding.

Source Data Discovery (SDD Projects)

The combination of Overlaps and Critical Data Element (CDE) Analysis and Unified Schema Builder provides a complete workbench for the analysis of multiple data sources and for prototyping the combination of those sources into a consolidated target, such as an MDM hub, a new application, or an enterprise data warehouse. These features help build unified data table schemas by accounting for known critical data elements and proposing statistically-based matching and conflict resolution rules before you have to write ETL code or configure an MDM hub.

The Overlap Analysis and Unified Schema Builder tools replace all of the manual work necessary when you combine data from multiple sources into a common target. The tools were inspired by customers who were accustomed to analyzing data sources individually, entering the data sources into a spreadsheet, and using that information to try to consolidate the data source.

Overlap Analysis and Unified Schema Builder replace all of the spreadsheet work by providing a data development environment where the analyst combines statistics and automatically discovered business logic with the ability to hypothesize and perform what-if analysis on the data sources.

The steps involved in these activities are shown below. The first three steps are Common Activities in All Projects. The next two steps are Overlaps and Critical Data Element (CDE) Analysis activities. The last two steps are performed using the Unified Schema Builder.
Overlaps and Critical Data Element (CDE) Analysis

IBM InfoSphere Discovery executes one-click overlap analysis on multiple data sources simultaneously. All columns are rapidly compared to all other columns for overlaps. The results are displayed in a graph and table format that allows you to drill down, view, sort, and filter the statistics. This automation makes performing overlap analysis on a large number of data sources extremely easy, allowing you to spend more time analyzing results and less time crafting SQL queries by hand.

The following Discovery screen shows the main Overlaps page. It summarizes the overlap results for three data sources: CRM, Community, and Region. All of the statistics are clickable links that drill down into the data and provide more detailed statistics.
The Overlaps step is performed just before you start using the Unified Schema Builder. In the Overlaps step, you compare data across all the sources you are evaluating. This analysis finds unique columns and redundant columns. The result is a fact-based measurement of the overlap strength between specific columns, based on the percentage of overlapping data values between the columns. This analysis is extremely valuable when you are consolidating multiple data sources or looking for sensitive data across a group of data sources.

In addition to identifying overlapping and unique columns, this analysis enables you to manage the process of tagging attributes that you consider critical to your analysis. These critical data elements, or CDEs, are the specific attributes that you want to include in your new target schema if you are migrating data or consolidating data into a new application, MDM hub or data warehouse.

Performing overlap analysis using CDEs helps you identify data sources which:

- contain many CDEs, which can often then be used to start constructing a unified schema that combines all of the sources
- are not overlapping
- subsume other data sources

You can also easily determine the level of consistency between overlapping data sources.
Unified Schema Builder

The IBM InfoSphere Discovery Unified Schema Builder uses an embedded workflow to help you complete your consolidation project on time and within budget. It uses the completed Overlaps analysis data as input for helping you determine the rules by which to consolidate data for various purposes, such as data migration, master data management or a data warehouse.

The following Discovery screen shows the Source Mapping page in the Unified Schema Builder. This page allows you to build SQL queries that extract data from multiple data sets into the target table.

Target Table Schema: Building Unified Data Models

Once you have tagged attributes as CDEs for inclusion in your new data model, you can start designing your new unified schema. You can also use untagged attributes.

There are three main approaches to creating a unified schema. Discovery supports all three approaches.

- Start with requirements, such as a set of reports and analytics to be supported by a data warehouse being modeled. Then construct a model that supports the requirements and map the CDEs to that model.
- Leverage the schema of one of your existing data sources, bringing just the CDEs into the new schema. Then, extend that model with CDEs from other sources.
• Use an existing industry schema, a purchased schema, or the schema of your target application (in the case of an application migration), then map the CDEs to that schema.

The first and second approaches are usually used together, with an analyst reworking a schema of the existing source to meet the requirements of the new application or data warehouse, and then extending it with CDEs from the other sources.

**Source Mapping & Unified Column Analysis**

Once you create the schema in the **Define Target Table Schemas** step, you map each data source to the unified schema in the **Perform Source Mapping** step.

Unified Schema Builder gives you powerful tools to help you develop your unified schema, including the ability to drag and drop elements from the data sources into your target schema and leverage the overlap information to match attributes from multiple sources automatically. You can also check for domain compatibility across combined attributes using unified pattern, length, and value frequency as well as complete column profile analysis for the combined attributes.

The following screen shows the pattern frequency for the Social Security Number column for each data source (CRM, Community and Region). The statistics for the union of those sources is shown in the Target column. Discovery found two patterns, one with dashes and one without. The data in the Community and CRM sources uses dashes, while the Region sources does not. Dashes will need to be added to the Region SSNs to be consistent when combined with the other data sources.

The Unified Schema Builder displays all of the target table's combined source data in a single view, allowing you to see the end result very early in the consolidation process. This initial prototype of your combined target happens long before you write any ETL or code your merge and match rules for an MDM Hub. As a result,
errors are caught early in the process and less rework is necessary later. As you work through the remaining steps, there are many opportunities to refine the initial prototype and add more detail.

**Matching Key Analysis**

One of the biggest challenges in combining multiple data sources is determining the *matching condition*, an expression involving matching key columns that align the rows across the various data sources. You may already know of a matching condition that is consistently used across all data sources, such as a transaction number, or the combination of First Name, Last Name, date of birth, address and a unique identifier (like a Social Security Number).

However, in most cases, such knowledge does not exist or has been lost. Different keys may be required to match rows from different sources and transformation expressions may be needed to match, normalize, and compare data. In these cases, you need to be able to rapidly prototype and iterate to determine the best matching condition.

It is commonly believed that MDM hubs resolve this problem. In truth, you must tell the MDM hub which columns to use as keys for matching and de-duplication.

Unified Schema Builder can automatically discover matching keys between any two data sources, even if the key is a composite key that involves many columns. You can rapidly prototype different matching keys to determine the best matching key across multiple data sources.

Unified Schema Builder then assesses the semantics of each matching condition using statistics and data views. The automated statistical analysis helps you determine if a particular matching condition results in over-matching (grouping apples and oranges together) or under-matching (placing similar apples in different groups). You can experiment with different or modified matching conditions and quickly view the results in an iterative process.

You use the resulting matching key information in your MDM hub or ETL process to align data from multiple sources when you populate your target.

**Refresh Unified Schema PF Key Statistics**

**About this task**

The **Re-run Step** button allows you to refresh the Unified Schema PF Keys statistics. Before performing this action, see **How to Use Unified Schema PF Keys**.

**Procedure**

1. Click **Re-Run Step** in **The Unified Schema PF Keys Window**. The **Processing Options Dialog** appears.
2. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   **Note:** Modifying task options is an advanced activity. Options are described in the Reference of Task Options. Modifying options and creating new option sets is described in Option Sets.

3. Click **Run** to refresh the Unified Schema PF Keys.

   **Note:** Although the Processing Options dialog closes immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the task status.

**Conflict Analysis:**

At this point in the process, you have created your new unified target schema, mapped source columns to the target, and identified a matching condition to align rows. The next step is to resolve any conflicts that occur due to different source values for the same target attribute.

One common approach is to determine the trustworthiness of each data source, assign precedence to each source for each attribute, and use the values from the source with the highest precedence. For example, when your own database, the CRM system, and the data warehouse have different values for “street address”, which source do you trust?
The answer often is based on the personal experience of the subject matter experts. This does not always accurately reflect the accuracy or consistency of the actual data. With Unified Schema Builder, however, you can now supplement this personal experience with cross-source data analysis that provides the following information for each attribute:

- Shows the number of sources mapped to each attribute.
- Determines the source that is the most consistent with other sources for each attribute.
- Automatically performs conflict detection using fuzzy and approximation matching (for example, dollar amounts are only in conflict if they differ by more than 2 cents).
- Automatically performs conflict resolution for each attribute based on the most recent value or the most trustworthy source.

Unified Schema Builder automatically generates trust rules assigned based on statistical analysis. You can still rely on personal experience in designating trust rules, but you now have solid statistical information to support what was formerly only gut feel analysis.

In the following screen, Unified Schema Builder has used statistical analysis to determine that CRM is the most trusted source for the SOCIAL attribute. The second most trusted source is Region, and Community is the third most trusted source.

Once this final analysis is performed, you now have all of the information you need to begin coding ETL, writing the migration scripts or programming your MDM hub merge/match rules. This methodology and automation process replaces what was previously a guessing game, significantly accelerating the cross-source analysis process, improving the quality of the results, and increasing the likelihood of success in your data consolidation projects.

**How to Use Discovery Studio**

Discovery is a complex and sophisticated product that contains many tools related to determining data types, discovering relationships between tables and columns, and discovering how to transform data to produce desired results. It also creates a variety of reports about the data.
Start Discovery Studio

About this task

Before starting Discovery Studio, make sure that the Discovery Engine Service and Discovery Server are configured and started as described in the IBM InfoSphere Discovery Installation Guide.

Start Discovery Studio by selecting Start>Programs>IBM InfoSphere>Discovery>Discovery Studio.

Specify a Discovery Server

About this task

When you start Discovery Studio, you are prompted to connect to a Discovery Server instance and a port. The values you enter become the defaults and are automatically populated in this dialog at future logins until you change them.

- **Server** is the name of the host of the Discovery Server you want to connect to. The default is LOCALHOST, which connects to a Discovery Server running on the same host as the Discovery Studio you are opening. Any server you enter here remains in the dropdown list for future connections.

- **Server Port** is 9090 by default. If you get a connection error using 9090, check with the person who installed Discovery Suite to see if it was installed on another port. (This is uncommon; more likely, the Discovery Server is not running.)

If you check **Automatically connect to this server**, Discovery Studio will connect to the specified Discovery Server and port without presenting this dialog at startup.

**Note:** You can open several instances of the same Discovery Studio application simultaneously on the same machine, each connected to the same Discovery Server or to different Discovery Servers.

Connect to a Different Discovery Server

To connect to a different Discovery Server when starting Discovery Studio, enter the hostname and server port of a different Discovery Server’s host. All of the projects in that Discovery Server’s repository are displayed in the Projects window.
About this task

To connect to a different Discovery Server during a Discovery Studio session, display the Home window and click the Server Connect button. This displays the Server Connection dialog and allows you to specify a different Discovery Server’s host and server port.

Exit Discovery Studio
About this task

You do not need to exit Discovery Studio to open or close projects.

To exit Discovery Studio, select Project>Exit Studio from the Project Menu in any Discovery Studio window. If a project has unsaved changes, you are prompted to save or discard the changes. Discovery Studio closes the project, and then Discovery Studio closes.

Discovery Server and the Discovery Engine Service remain running. Any queued tasks remain in the Discovery Engine queue and are executed as soon as a Discovery Engine becomes available. The next time you open Discovery Studio, if the task has finished, you can view the results.

Order of Steps

Discovery Studio has a series of tabs across the top of the screen. Some of the tabs are different for SDD and TD projects. Each tab represents a step, and the steps are performed in order, left to right. In each step, you discover, define, or review data relevant to your project.

Because of the many different kinds of discovery tasks that can be accomplished with Discovery, you don’t always need to perform every step in Discovery Studio. When appropriate for your project, you can Skip Steps after creating a project and importing tables and text files. Depending on the goal of a particular project, you can skip discovering or defining data in various steps whose results are not required for that particular goal. However, always work from left to right in the tabs.

Note: Perform Column Analysis is always recommended unless you are confident there are no VARCHAR strings containing DateTime or numeric values in the data.

In some situations, you may need to repeat previous steps. For example, if you perform discovery all the way up to Data Objects and then Edit a PF Key, you should Re-Run PF Keys and then Re-run Data Objects. Note that when you repeat steps, you still perform them in left-to-right order.

A good way to determine which steps are necessary for your project is to read the IBM InfoSphere Discovery Methodology Guide and use the IBM InfoSphere Discovery Quick Reference. Both provide guidance in using Discovery to achieve various goals.

Steps in TD Projects

Transformation Discovery Projects contain seven tabs The last tab, Maps, displays six sub-tabs that are also performed in left-to-right order. Depending on the data in the mapping project and the goal, you may not need to perform all Maps steps.
Steps in SDD Projects

Source Data Discovery Projects contain seven tabs. The last tab, Unified Schema, contains four sub-tabs. All tabs are executed in left-to-right order. Depending on the goal of the SDD project, you may not need to perform all SDD steps.

Move from One Window to the Next

After completing work in each tab, you continue work in one of the following ways:

- Click the Run Next Steps Button to perform discovery for the next step. You can also Execute Multiple Steps at the Same Time.
- Click the Next Window’s Tab
- Re-run Step if you made changes that require re-running the step.
- Skip Steps and execute a step that is several tabs ahead of the current one.

Note: Whenever any task runs, it relies on staging tables stored in the staging data source. Do not make any changes in the staging data source (drop or reload any tables) while any task is running. If you do, the results may not be correct.

Although Discovery Studio immediately displays the next tab when you execute a task, the actual data processing for the new tab may not yet be finished. While data is being processed, a message displays onscreen indicating the project is locked, and the Activity Viewer status shows one or more active tasks. These messages disappear when processing is complete and you can continue work.
Click the Run Next Steps Button
About this task

The standard way to proceed to the next window is by clicking the Run Next Steps button in the lower right corner of the current window. This displays the Processing Options Dialog, allowing you to specify thresholds and execute the tasks necessary to populate the next window.

By default, the tasks are performed on all tables and columns in the source and target data sets. Some steps allow you to select a data set or specific tables or columns for processing.

Note: After you queue tasks, you can close the project and open a different project, or close Discovery Studio. See The Active Window (Scheduling Task Execution) and How to Execute Tasks for information about how queued tasks are executed.

Click the Next Window’s Tab
Another way to move to the next step is to click the next step’s tab. This opens the window but Discovery does not perform any discovery tasks, so the window is either unpopulated or contains only information carried over from the previous step.
About this task

You can use this method when you want to skip steps or advance to a particular step without having Discovery perform discovery first, possibly because you want to manually enter the data.

For example, suppose you want to manually create PF Keys. To do this:

Procedure

1. Create a project in the Home tab. When you click OK, the Data Sets tab automatically opens.
2. In the Data Sets tab, populate the data sets.
3. In the Data Sets tab, click Run Next Steps and select Column Analysis, then click Run.
   Discovery displays the Column Analysis tab and performs column analysis on the data sets.
4. In the Column Analysis tab, after reviewing the Column Analysis results, click the PF Keys tab instead of clicking Run Next Steps.
   Discovery opens the PF Keys window with a blank diagram pane. There are no PF Keys listed because that task has not yet been executed. (However, any PF Keys that were defined in metadata are shown.) You can now manually create the PF Keys in this window.
   Alternatively, you can now click Re-run Step in the PF Keys tab to have Discovery perform PF Key processing.

Re-run Step
About this task

The Re-Run Step button allows you to run the discovery actions for the current window. For example, clicking Re-Run Step in the Column Analysis window repeats the Column Analysis discovery tasks and refreshes the Column Analysis window with the new results.

If you displayed a step by clicking on the step’s tab instead of using Run Next Steps, the Re-Run Step button causes Discovery to perform discovery for the displayed step.

Re-running a step sometimes allows you to select one or more specific processing actions that are not available or cannot be performed individually by selecting Run Next Steps from the previous tab. For example, if you are in the Column Analysis window and click Re-Run Step, you can select Only Stage Sample Sets or Only Refresh Statistics. These options are not available when you click Run Next Steps from the Data Sets tab.
Skip Steps
Several factors are involved in deciding whether or not to skip a step:

About this task
- Does your project require the results from the step?
- Do any subsequent Discovery steps or processing actions rely on the results from the step?
- If Discovery requires the results to run discovery in a subsequent step, do you know the data well enough to skip discovery in that later step and define those results instead of discovering them?

To skip a step, simply click on the tab of the step you want to work in next. (Remember to work left-to-right.) The step displays without any discovery being performed, but the screen may contain incomplete or inaccurate data. If the screen is empty, you may not be able to perform certain actions in the window.

For example, if you are in Column Analysis and want to skip the PF Keys step, click on the Data Objects tab. You can now define data objects, or click Re-run Step to perform data object discovery. Alternatively, you can display the PF Keys step and click Run Next Steps to perform data object discovery.

However, some later discovery tasks require information from previous screens. For example, in TD projects, join conditions are based on PF Keys. You can skip PF Key discovery if either one of the following conditions is true:
- You already know that none of your target columns are derived from joined source tables.
- The project needs join conditions but you already know what they are. In this case, you will define the joins after defining the maps.

But, if you are not sure whether the project uses join conditions, or if you do not know what they are, you need to either run PF Key discovery or define PF Keys so that Discovery can create joins when generating maps.

Another example is target matches. Discovery needs to know what the target matches are before it can discover maps. If there aren’t any Target Matches, the Discover Maps step fails. However, if you already know which source tables map to which target tables, you don’t need to run Target Matches or Discover Maps.

After reviewing the Data Objects results, you can skip right to the Maps Summary tab and manually define the maps. In a large project, this can save time by not needing to run the Target Matches and Maps processing steps.

Overview of Discovery Studio Steps
Because of the many ways in which you can use Discovery, there is no single set of instructions that will serve all users.

Many types of data discovery projects can use the tools that Discovery provides to quickly and accurately provide the desired results, and not every project will use Discovery’s full capabilities. For example, you can use Discovery to:
- Determine the data types of all data within a data set.
- Discover primary key-foreign key relationships between tables.
- Discover sensitive data wherever it appears in a data set. You can generate a report of that data, and also use Discovery to create a modified data set that masks all sensitive data.
• Discover matching source and target data.
• Discover transformations that, when applied to source columns, accurately transform the data to match the data in target columns.
• Create relevant sample sets for various non-mapping related purposes.

**Common Steps for All Projects**

The following steps are common to both SDD and TD projects, although the screen shots may show one or the other type of project.

**Create a Project**

**About this task**

The *Home* window is the starting point in using Discovery.

When you first open Discovery Studio, you are in the *Home Tab* with the list of Source Data Discovery projects displayed by default. The remaining tabs in Discovery Studio are not displayed until you create and open a project.

All processing and analysis tasks for a particular group of user data files are performed in the context of a project. You will probably create a number of **Source Data Discovery Projects** and **Transformation Discovery Projects**, each one using different data sets or solving a different business problem.

**What to Do in This Step**

If you expect several projects to be using the same staging data source, use **Tools>Options> Default Staging Data Source** to create a default staging data source. This makes it unnecessary to enter the same staging data source information for each project.
Click **New Project** to start creating a project. Give it a name, and specify the default staging data source or a different one. You can also specify a password required to open the project, and enter a description of the project.

At any time in the mapping process, you can **Edit a Project’s Properties**, including changing the project name or description, as well as change, add, or remove the password.

You can **Connect to a Different Discovery Server** in this window at any time. The Project List will display the projects in the new Discovery Server’s repository.

Proceeding to the Data Sets Window

After you create a project, Discovery Studio automatically opens the **Data Sets** window. You can also click the Data Sets tab to open it.

**Create Data Sets**

**About this task**

A data set is a collection of database tables and text files to be processed, analyzed, or mapped. It can contain as many database tables from as many database types and JDBC connections as needed, and as many delimited or positional text files as needed.

SDD projects use at least one data set. You can create as many additional data sets as needed.

TD projects contain a **source data set** and a **target data set**, as shown above, with an optional data rule dataset (see **Generate Data Rules**). If your goal in a TD project is not to discover transformations but only to discover relationships within a single data set, you can leave one data set empty.

**What to Do in This Step**

- **Name and Classify the Data Set** and then **Import Database Tables** and **Import Text Files** as needed. You can **Modify a Logical Table**, **Define a Logical Table**, and **Define a Sample Set** now or later, if needed. You can return to this window at any time and **Modify Physical Tables in the Data Set** including adding or deleting...
tables and data rule tables. If you change any physical tables, logical tables, or sample sets, re-run all discovery tasks to obtain new results.

In TD projects, if you are importing data rules, you can Import Lookup Tables now, but you can also return to this screen and import them later. If you do, it is not necessary to repeat any discovery tasks that do not reference the data rules.

Proceeding to the Column Analysis Window

Clicking Run Next Steps in the Data Sets window opens a dialog that allows you to Perform Column Analysis.

Perform Column Analysis

Some database tables contain correct data type definitions in metadata. However, other database tables contain incorrect or incomplete metadata, and text files do not define the data types.

About this task

In this step, Discovery calculates and displays statistics about the data in the data sets, along with displaying the metadata information. Discovery also examines all VARCHAR strings to determine if they contain date/time or numeric values, and changes them to the appropriate data type.

What to Do in This Step

After you Perform Column Analysis, you need to Review Column Analysis Data. Verify the data types and make any necessary modifications, such as changing the length of a column or correcting a wrong data type defined in the metadata. Mark columns that are important to your project as Critical Data Elements (CDEs).
Whenever you modify the tables in a data set, you need to Re-run Column Analysis.

Proceeding to the Next Step

Clicking Run Next Steps in the Column Analysis window opens a dialog that allows you to Discover PF Keys.

Discover PF Keys
About this task

Discovery automatically imports PF Key relationships when they are defined in a table's metadata (and if import is supported by the JDBC driver). When the relationships are not defined, Discovery discovers column matches by examining the actual data. Column matches with the highest hit rates and selectivity are automatically designated as PF Keys.

The diagram in this window displays the tables in each data source that have a column relationship. Primary-foreign key relationships are represented by solid lines, and column matches are shown as dashed lines. Selecting a line displays the expression in the Relationship pane below the diagram.

What to Do in This Step

Review PF Keys by examining the statistics in the PF Keys Grid - Foreign Keys and the PF Keys Grid - Column Matches. Use the Data Preview screens to show hits, misses, and duplicates. Define, modify, or delete column matches and PF Keys.
Keys. In order to verify the accuracy of the results, you may need to discuss them with other data analysts or research existing records.

If you define or modify discovered column matches or PF Keys, Re-Run PF Keys.

Proceeding to the Next Step

Clicking Run Next Steps in the PF Keys window opens a dialog that allows you to Discover Data Objects.

Discover Data Objects

About this task

A data object is a conceptual way of looking at table relationships within a data set. A data object represents a group of tables that are related by PF Keys.

A data set can contain many data objects, with each data object consisting of many tables or just one table. A table can be both a parent and a child, so the same table may appear in several data objects.

This window displays the data objects in each data set, with the join conditions represented as solid lines. Click on a line to display the join expression in the Relationship pane below the diagram.

What to Do in This Step
Review Data Objects. You can Define Data Objects or delete data objects, and Add Child Tables or delete them from data objects. Also in this screen you can Create a Related Sample Set from a data object.

Proceeding to the Next Step

Clicking Run Next Steps in the Data Objects window opens a dialog that allows you to Discover Target Matches.

SDD Projects Only

About this task

These tabs are only available in Source Data Discovery Projects.

Discover Overlaps

The Overlaps step provides a clear picture of overlapping data in your sources.

About this task

What to Do in This Step

Review the column data to verify that the discovered overlaps are useful and valid. Delete any obvious mismatches in the Value Overlap Details. If there is any
If you have any doubt about the data in a particular overlap, use Column Summary and Column Overlaps to display the actual data. Define new overlaps and have Discovery determine the statistics.

Mark columns that are important to your project as Critical Data Elements (CDEs).

Proceeding to the Next Step

Overlaps discovery is now complete. You can click Unified Schema to begin defining a unified schema.

**Create a Target Table for a Unified Schema**
This tab is only available in SDD projects.

**About this task**

The Unified Schema tabs help you consolidate and de-duplicate data from different data sources into a target table.

**What to Do in This Step**

Define the target table schema. Populate the table with as many target columns as needed. You can modify data types and rearrange the column sequence.

Proceeding to the Next Step

Click the Source Mapping tab to define the source mapping for these columns.
Define Source Mapping
This tab is only available in SDD projects.

About this task
In the Source Mapping Window, you define the SQL (select, join, where, group by, having) that Discovery uses to extract data from each data set into the target table. The target table can contain records from any or all data sets in the project.

What to Do in This Step
Map table columns to target columns and create filters for each table column until each target column contains appropriate data from the relevant tables. If necessary, return to the Target Table Schema tab to refine the target column characteristics.

Proceeding to the Next Step
Clicking Run Next Steps in the Source Mapping window opens a dialog that allows you to Run Unified Column Analysis.

Perform Unified Column Analysis
This tab is only available in SDD projects.

About this task
In the Unified Column Analysis Window, you consolidate data from different sources into the aggregated target table. The statistics and data previews in this screen are used to define matching conditions for use in the next step, Match and Merge.
What to Do in This Step

Review the resulting statistics. Statistics are shown for the aggregated columns in the target table, as well as for each individual table column used in the target table.

As needed, refine the target table by returning to the Target Table Schema window and adding or removing target table columns. You may also change the source mapping.

Proceeding to the Next Step

Clicking Run Next Steps in the Unified Column Analysis window opens a dialog that allows you to Run Match and Merge Analysis.

Perform Match and Merge Analysis

This tab is only available in SDD projects.

About this task

The Match and Merge Analysis Window contains two separate processes performed on the target table schema.

Procedure

1. In Matching, you define matching conditions to split the aggregated target table records into groups containing matching (duplicate) records.
2. In Merging, you build a unified target table in which each record represents one group (with the best value selected from the alternatives) from the aggregated target table.

**Results**

What to Do in This Step

Define matching conditions for each target column. Also define conflict detection rules that determine whether the values in each group are considered conflicts or not. Merge the results and define conflict resolution rules that select the best value from the duplicate record alternatives.

Proceeding to the Next Step

The unified schema is finished. You can [Generate Reports](#) and [Export Data](#) to import into metadata repositories or ETL tools.

If you want to discover PF Keys for the unified schema, click the **Unified Schema PF Keys** tab.

**Discover Unified Schema PF Keys**

This tab is only available in SDD projects.

**About this task**

After creating or importing at least two unified schema target tables and running all Unified Schema steps, you can discover and analyze PF keys within the target tables.
At least two target tables must exist, with at least one target column each.

What to Do in This Step

Define PF Keys for the unified schema, or import a CWM file containing PF Keys. Refresh the statistics and review the results, making changes as necessary to improve the results.

Proceeding to the Next Step

This is the last step in an SDD project. You can Generate Reports and Export Data.

**TD Projects Only**

**About this task**

Transformation Discovery Projects allow you to discover cross-source transformations.

The Maps tab contains six sub-tabs that allow you to view, modify, and refine the results of the Discover Maps discovery task.

Work left-to-right in the Maps steps, skipping any steps that do not apply to the map. For example, if your map does not use Where clauses, you can proceed directly from Bindings to Transformations; however, do not create Reverse Pivots before the Transformations are finished.

The first sub-tab, The Summary Window, displays a high-level summary of the results of the Maps processing task.

The remaining sub-tabs allow you to modify and refine the results to improve the source-to-target transformations.
Common Elements in the Maps Sub-tabs are described below. Specific information about each sub-tab is available in each section:

- Discover Joins
- Discover Binding Conditions
- Discover Where Clause
- Discover Transformations
- Discover Reverse Pivots

**Common Elements in the Maps Sub-tabs**

All Maps sub-tabs are displayed in the center pane of the Maps tab. The Maps sub-tabs share a number of common elements. These are described below.
The Maps List

About this task

The Maps list displays all discovered and defined maps in the project. The same Maps list is shown in each Maps sub-tab; it does not change depending on the Maps sub-task that is displayed. Using Autohide Pane, you can minimize the Maps list to increase the viewing area of the Maps sub-tabs.

The list contains an entry for each target table. Under each target table in the list, the discovered maps are shown (more than one map can be generated for each target table). A map always has one target table (see Determining Which Data Set is the Target), but can have one or more source tables.

When no map is selected, you cannot display any Maps tab except the Summary tab. Select a map in the list to display its data in the Maps sub-tabs.
Select a map (not a target table) to view SQL expressions, statistics, and other information about that map in the Maps sub-tabs.

When you select a map, Discovery Studio automatically opens the sub-tab that was most recently active for that map.

**Approval Checkboxes**

*About this task*

Approving a map is a convenient way to indicate the maps you have reviewed or verified.

A map’s approval status does not have any impact on subsequent data discovery performed in the Maps tab, or when re-running previous steps.

**Delete a Map**

To delete a map, select it in the Maps list and click X.

**Maps Approval Checkboxes**

*About this task*

Maps have approval checkboxes. You can use these to indicate maps you have finished reviewing.

**Run Next Steps and Re-Run Steps**

*About this task*

Most maps sub-tabs have their own Run Next Steps and Re-Run Steps buttons. (The first sub-tab, Summary, does not have a Re-Run Step button, and the last one, Reverse Pivots, does not have Run Next Steps.) Generally, you only execute one step at a time in the Maps sub-tabs.

Note: Work left-to-right in the Maps steps, skipping any steps that do not apply to the map.
The Run Next Steps buttons allow you to repeat some of the discovery actions performed during Discover Maps after you have made changes to the discovered results or defined new elements. You can also perform discovery that is not normally included when you generate maps, like Discover Pivot Columns.

If you skip a step, you can execute the next step in two ways: by using Re-Run Step, or by opening the skipped step and clicking Run Next Steps. For example, if you go directly from Bindings to Transformations (skipping Where Clause), you can execute Transformations discovery by using the Transformations Re-Run Step, or by opening the Where Clause tab and clicking Run Next Steps.

The Re-Run Step buttons allow you to fine-tune the maps by providing additional options that are not available in the Run Next Steps task. The specific options are described in that sub-tab's section.

Only Refresh Statistics

About this task

You can refresh all statistics in a Maps sub-tab by re-running the step using the Only Refresh Statistics option. Each Maps sub-tab has this option. It allows you to select some or all source and target columns, and then refreshes the statistics for those columns in the selected tab. A Discovery Engine executes this task in the background, allowing you to continue working in Discovery Studio.

(By comparison, the Refresh Statistics icon refreshes the statistics only for the selected expression, and it executes fairly quickly.)

You may need to refresh statistics for all subsequent Maps sub-tabs after changing an expression that many subsequent expressions are based on, such as a join or a binding condition. Check subsequent Maps sub-tabs after making a change to see if any statistics have become stale. Because refreshing all statistics can be very time-consuming, try to make as many changes as possible in a tab before refreshing all statistics.

Note: Changes to expressions only affect subsequent sub-tabs, not previous ones. For example, if you change a Where clause, you need to refresh statistics for all transformations; but when you change a transformation, the Where Clauses do not become stale.

To refresh statistics for some or all of the source and target columns in the current Maps sub-tab:

Procedure

1. Click Re-Run Step in one of the Maps tab sub-tasks.
2. In the Processing Options dialog, check Only Refresh Statistics. This may be the only option available, or there may be others, depending on the particular Maps sub-task.

Note: When Only Refresh Statistics is selected, you cannot select any other options.
3. To refresh statistics for all columns, click **Run**.

4. To refresh statistics for only certain columns, click **Select** to activate the tables and columns listed in **The Parameters Pane**. Check the tables and columns that you want to refresh statistics for, then click **Run**.

**Results**

Discovery queues the task and locks the map, and a Discovery Engine picks up and executes the task. If there are many columns selected, it may take some time to complete it.

In the meantime, you can continue working in Discovery Studio. The map is locked for writing, but you can still work in it in view-only mode until the task completes. Alternatively, you can work on a different map, open a different project, or **Exit Discovery Studio**. The Discovery Engine continues working on the task until it is complete.

**Generate Target Matches**

This tab is only available in TD projects.

**About this task**

**Target matches** define a relationship between one source column and one target column. Discovery discovers target matches by analyzing and comparing the actual data within the source and target data sets. Each related source and target table has at least one target match, and may have several. Target matches are displayed by clicking **Show All**.

**Target maps** are visual representations of target matches. The **Target Matches List** **Target Matches Diagram** and **Target Matches Grid** display the target maps.
What to Do in This Step

**Review Target Matches**  You can **Define a Target Map** and **Define a Target Match**.

Instead of using the **Discover Target Matches** task to create target matches and target maps, you can jump to this screen directly from The Column Analysis Window and use **Re-run Target Matches** to generate target matches only. Target maps are not displayed, so this screen will appear empty, but you can display the target matches by clicking **Show All**. This allows you to **Discover Maps** and discover transformations.

Proceeding to the Next Step

Clicking **Run Next Steps** in the Target Matches window opens a dialog that allows you to **Discover Maps**.

If you are manually defining maps, do not use **Run Next Steps**, but instead click directly on the Maps tab and display the Summary screen.

As necessary, refine the results and run or re-run the Maps discovery tasks to obtain the best possible transformations.

**Discover Maps**

**About this task**

This tab is only available in TD projects.
Discovery automatically generates maps containing one target table and the source table(s) that map to it. The map contains the join conditions, binding conditions, and transformations that are applied to the source data in order to create the target data, on a column-by-column basis.

What to Do in This Step

Review the summary information for each discovered map, verifying that the correct source table(s) and target table are included. Then, for each map, work left-to-right in the Defining a Map as described in the following pages, skipping any steps that do not apply to that map. For example, if your map does not use Where clauses, you can proceed directly from Bindings to Transformations; however, do not create Reverse Pivots before the Transformations are finished.

Alternatively, you can Define a Map manually if you already know certain information, such as any join conditions.

Proceeding to the Next Step
If Discovery discovered the maps and any of the maps have more than one source table, click the **Joins** tab to review those maps' joins. If none of the maps have joins, skip Joins and click the **Bindings** tab.

If you are manually defining a map with more than one source table, click **Run Next Steps** to have Discovery find joins, or just click the **Joins** tab to manually define the join.

If you are manually defining a map that has only one source table, skip Joins and click the **Bindings** tab.

**Discover Joins**
This tab is only available in TD projects.

**About this task**

This step is only necessary for maps containing more than one source table. Each map in your project may have a different number of source tables.

[Image of the user interface with the Joins and Bindings tabs highlighted.]

**What to Do in This Step**

If Discovery discovered the joins, **Review Joins** to determine if each map's discovered joins are appropriate and accurate.

If you are manually defining a map with two or more source tables and want Discovery to find the join, **Generate Joins**. Review the results.

If you are manually defining a map with two or more source tables and know the join condition already, **Define a Join** and refresh its statistics, then review the results.

**Proceeding to the Next Step**

If Discovery discovered the maps, click the **Bindings** tab to review each map's binding condition.

If you are manually defining a map, click **Run Next Steps** to open a dialog that allows you to discover the binding condition, or just click the **Bindings** tab to manually define the binding condition.
Discover Binding Conditions
This tab is only available in TD projects.

About this task
Each map has a binding condition between the target table and the source table(s).

Review Binding Conditions, Generate Where Clauses if needed, Review Transformations and Discover Pivot Columns and Generate Reverse Pivot Maps if needed.

What to Do in This Step
If Discovery discovered the bindings, Review Binding Conditions to determine if each map’s binding condition is appropriate and accurate.

If you are manually defining a map and want Discovery to find the binding condition, Generate Binding Conditions. Review the results.

If you are manually defining a map and know the binding condition already, Define a Binding Condition and refresh its statistics, then review the results.

Proceeding to the Next Step
If a map needs a where clause, click Run Next Steps for that map to open a dialog that allows you to Discover Where Clause. You can also click the Where Clause tab to Define a Where Clause.

If the map does not require a where clause, click Run Next Steps to open a dialog that allows you to discover transformations.

Discover Transformations
This tab is only available in TD projects.
About this task

The goal of TD projects is to discover expressions that transform the source data into the target data.

Procedure

1. If Discovery discovered the transformations, Review Transformations to determine if each transformation is appropriate and accurate.
2. Edit Transformations and change the Primary Transformation if necessary.
3. If you are manually defining a map and want Discovery to find transformations, then Discover Transformations, and review the results.
4. If the map needs data rules, Generate a Data Rule From the Transformations Tab. You can also Define a Data Rule Using an Imported Lookup Table, or you can Create a Data Rule by Defining and Populating a Table.
5. If a map needs Reverse Pivots, click Run Next Steps to open a dialog that allows you to Define a Pivot Column or Source Column for that map.
6. When you have finished reviewing the results, you have completed all of Discovery's discovery, analysis, and mapping tasks.
7. You can Show Map SQL to view the map as a single complex SQL statement, Generate Reports to view the results of the mapping exercise in different formats, and Export Data to import into metadata repositories or ETL tools.
8. If you are using Exception Manager, you can now Create Validation Jobs (TD projects only).

How to Execute Tasks

The Processing Options Dialog

About this task

This section describes features common to Processing Options dialogs. Some features are available only in TD projects or only in SDD projects.
You can also view general information about Move from One Window to the Next.

Or, to use the Processing Options dialog for specific tasks, see the following:

- Perform Column Analysis
- Discover PF Keys
- Discover Data Objects
- Run Unified Column Analysis
- Run Match and Merge Analysis
- Refresh Unified Schema PF Key Statistics
- Discover Target Matches
- Discover Maps
- Generate Joins
- Generate Binding Conditions
- Generate Where Clauses
- Discover Transformations
- Discover Pivot Columns

Clicking Run Next Steps displays a Processing Options dialog containing options for executing the next step. This dialog is similar for every step; the only differences are whether you can select specific objects for processing, and the list of options used in the step.
When you click Run, Discovery queues the selected step, opens the next window, and populates it with the results. It may take several minutes or longer for the task to actually be completed.

The Parameters Pane

By default, discovery is performed on the full content of the source and target data sets. However, some steps allow you to select the data set or the specific tables or columns to perform discovery on. In the Maps tab, the selected map determines which tables and columns are shown here.

About this task

When discovery can only be performed on the full content of both data sets, the Parameters pane is empty. Tasks that allow content to be selected show the items as grey text.
This task is performed on the full content of both data sets. You cannot select specific content for discovery.

This task allows you to select specific content for discovery.
To run the discovery task on certain content only, click the **Select** option. The list becomes active with all items checked. Check or uncheck the items as appropriate.

**The Activity Pane**

**About this task**

The Activity pane displays the name of the task about to be executed, along with the name of the step you are currently working in.
The **Label** is a name for this specific execution. This is the name shown in the **Activity Viewer** to identify the execution.

By default, the label uses the current date and time. You can change the label in the **Processing Options** dialog by clicking in the **Label** field and typing a new name. The name does not need to be unique.

### View Options Window

#### About this task

To view the options available in a step, display the **Processing Options Dialog**. If the Default Option Set is selected, click **View** to open it in read-only mode (because the default option set cannot be edited). The options for each step are grouped by type and show the default values.

To create a new option set, click **New**.

Click on an option name to display a brief description. All options in each step are described in the corresponding chapter of the documentation.

You can drag the column border to view the full option names. You can also drag the border of the dialog to change its size.

You can sort the options in the list alphabetically or view by group by clicking on the appropriate icon above the list.

You can also view the option sets of other windows by clicking the **Step** field and selecting a different window from the list. When the current step is a non-Maps...
step, the non-Maps steps are shown; when the current step is a Maps step, only
the Maps steps are shown.

Create a New Option Set

About this task

You cannot edit the values in a default option set. To change any values, create a
new option set and edit the values in it. You can create as many new option sets as
necessary.

Note: New option sets are available only in the project they were created in. You
can [Export and Import Option Sets] into different projects.

New option sets are populated with the default values, which you change as
necessary. You can change just one option in one step or modify options in many
steps to fine-tune the option set for a particular purpose, such as stricter or looser
thresholds, profiling in parallel, etc. Once you change a value it is shown in bold
text to indicate it has been modified.

When you click New in the Processing Options dialog, a new option set is created.
The option values are set at the defaults.

You can change default values for one or more screens and save the changes as a
new option set.

To change one or more values in an option set:

Procedure

1. In [The Processing Options Dialog] select the option set that you want to use as
the basis for the new option set. This can be the default option set or one of the
[Custom Option Sets].
2. Click New. The New Options popup appears with a copy of the selected option
set named NewOptionSet.
3. In the Name field, rename the option set. Use a descriptive name that clearly
identifies the purpose of the custom option set.
4. Change the option values as desired. Non-numeric options have a dropdown
menu listing the available settings. Numeric options sometimes have valid
ranges. See the Options table in each step's chapter in this document for
detailed information about each option and how changing the default value can
affect the results.

After a setting has been changed, it appears in bold. This makes it easy to
identify the changed settings in an option set.
5. If needed, modify the options for additional steps by clicking in the Step field
and choosing the appropriate step name. The options for that step are shown in
the New Options window and you can change them as described above. You
can also change these options later when you run other steps.
6. When you are finished changing options, click OK. The new option set is
saved. You can now select this option set in any step in this project and
Discovery will use these settings to execute the steps.

Note: If you are not going to execute the task right now, you can click Cancel
to close the Processing Options dialog. The new option set is still saved.

Edit an Option Set:
All of the option sets provided with Discovery are read-only and cannot be changed or deleted. To edit the option values in a set, [Create a New Option Set](#).

You can edit any option set you have created by selecting it from the list in the Processing Options dialog and clicking **Edit**.

**Export and Import Option Sets:**

Option sets are specific to a particular project, but you can export and import option sets from one TD project into another TD project.

**About this task**

To export an option set (TD projects only):

**Procedure**

1. Open the project containing the option set.
2. Select **Project>Export>Project Objects**.

**Results**

The *Export Objects* dialog appears. All option sets in the project are listed.
1. Select the option set(s) to be exported.
2. Click **Browse** to select the destination directory, then click **OK**. A folder with the project name is created in the specified location, and the selected option sets are saved in the folder as XML files.

To import an option set (TD projects only):

1. Open a project.
2. Select **Project>Import>Project Objects**. An import dialog appears.
3. Select the folder containing the exported option set, then click **OK**. (The contents of the folder are not displayed.) The *Object Import* dialog appears.
4. Shift-click or Ctrl-click to select the desired option sets in the list and then click **OK**. The option sets are imported into the project.

**The Steps Pane**

**About this task**

In the **Processing Options Dialog**, you have the option of executing only the next step, several of the remaining steps, or all remaining steps.

- If you select only the next step, Discovery executes only the tasks related to that step and opens the next window.
- If you select two or more steps, Discovery executes the tasks involved in all of those steps, in order, and opens the window corresponding to the last step selected.

**Execute Multiple Steps at the Same Time:**

**About this task**

**Note:** You can [Edit an Option Set](#) for one or more of the additional tasks. When you click **Run**, each step will use the option set selected for it.
To execute more than one step as one task, drag the arrow down to the desired step. All steps between will be executed also; you cannot skip steps when selecting multiple steps.

**Locked Objects**

During task execution, the entire project or just the affected table or map can be locked. You can work in other projects or objects when a particular project or object is locked, as long as the new work does not involve any locked objects.

- When a task is run or re-run in the Data Sets, Column Analysis, PF Keys, Data Objects, or Target Matches tabs, Discovery locks the entire project. You can open another project during this time and work in it.
- When a task is run or re-run in any of the Maps tabs, Discovery locks only that map. You can work in another project or another map during this time.

**Note:** If you selected only certain tables, maps, or other objects for processing, only they are locked. You can work on the other tables, maps, or objects.

A message in the Discovery Studio interface indicates when a map or project is locked. In the screen below, a task for the first map is running so the first map is locked. The other two maps are not locked.

Locking prevents conflicts that may occur from the same object (table, map, or other item) being manipulated by more than one task at the same time.

The following table lists the objects that are locked during various discovery tasks.

<table>
<thead>
<tr>
<th>Discovery Action</th>
<th>Source Objects Locked</th>
<th>Target Objects Locked</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Analysis</td>
<td>Source data set, Source data set Tables</td>
<td>Target data set, Target data set Tables</td>
</tr>
<tr>
<td>PF Keys</td>
<td>Source data set</td>
<td>Target data set</td>
</tr>
<tr>
<td>Data Objects</td>
<td>Source data set, Source data source Tables, Source data objects</td>
<td>Target data set, Target data set Tables, Target data objects</td>
</tr>
<tr>
<td>Target Matches</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Activity Viewer

**About this task**

The Activity Viewer displays the current activities and a history of previous activities performed in this project. Before you queue any tasks, this window is empty.

To open the Activity Viewer, click the Activity Viewer status link in the upper right corner of the Discovery Studio screen.

The Activity Viewer for a TD project is shown below. The Activity Viewer for SDD projects does not have a **Map Activities** tab.
For each individual task run, the Activity Viewer indicates whether it was successful, completed, cancelled, failed, and other information. You can view Trace, Error, and Debug Log Files for each task as well as the options that were used to execute that task.

You can also right-click in a Project Activities or Map Activities column header to display the Column Chooser.

**Project Activities and Map Activities**

The Activity Viewer lists all tasks that have been queued in this project. For ease of use, tasks are divided into two tabs: Project Activities and Map Activities.

The Project Activities tab include all tasks you execute in the main tabs, like Column Analysis and PF Keys Parent- Classification.

**Note:** In Transformation Discovery Projects, the Activity Viewer contains a Project Activities tab and a Map Activities tab. In Source Data Discovery Projects, there is only a Project Activities tab.
The **Map Activities** tab, shown only in Transformation Discovery projects, includes all tasks you execute in the **Maps** sub-tabs, like **Joins** and **Bindings**. This tab also contains an **Object** dropdown menu.

<table>
<thead>
<tr>
<th>Label</th>
<th>Start Time</th>
<th>End Time</th>
<th>PID</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Analysis</td>
<td>12/28/2006 7:02 PM</td>
<td>12/28/2006 7:03 PM</td>
<td>1000-cc27...</td>
<td>completed</td>
</tr>
<tr>
<td>Column Analysis</td>
<td>12/29/2006 2:34 PM</td>
<td>12/29/2006 2:35 PM</td>
<td>1092-d059...</td>
<td>completed</td>
</tr>
<tr>
<td>Column Analysis Statistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Column Matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Activity, Status, and Object Filters

About this task

The filters in the upper pane of the Activity Viewer change depending on the tab selected. The Map Activities tab contains an additional filter, Object.

You can filter tasks in both tabs by Activity and Status. These filters can be used individually or in combination on each tab.

The Activity filter shows only tasks of the selected type in the display.

- When Project Activities is selected, the Activity filter list contains each task available in The Processing Options Dialog when you click Run Next Steps or Rerun Steps in any of the main tabs. This group includes the Maps task executed in the Target Matches tab.
- When Map Activities is selected, the Activity filter list contains the tasks available when you click Run Next Steps or Rerun Steps in any of the Maps sub-tabs.
The **Status** filter display only tasks with the selected status.
The following screen shot shows the results under two steps, Classifications and Column Analysis. One task has been run under each step. In the Classifications list, the Messages tab is selected. In the Column Analysis list, the Options tab is selected, displaying the thresholds used in this particular task.
Changing Project Settings During a Task

When a task is scheduled or is running, the project is locked. You cannot perform some actions while the project is locked, such as selecting a new primary sample set, changing a column’s data type, adding a new PF Key, and so on.

After processing is finished, the project is unlocked and you can make changes to it.
View Task Progress
About this task

While a Maps task is running, a status bar gives an approximate report of the task’s progress. This is convenient for estimating the progress of long-running tasks.

Note: The status bar is not available in non-Maps tasks.

The status bar can be turned off and on using the Enable discovery progress indicator option. When running, the status bar is updated at intervals determined by the option Minimum amount of time between DPI updates.

View Task Activity Summary
About this task

You can view a brief summary of the activities performed during a particular task.

Procedure
1. Display the Activity Viewer.
2. In the Activity list, click on the expand button for the desired task.
   • Project Activities are grouped by tab. There is an entry in the list for each time that particular task was queued.
   • Map Activities are grouped by map. There is an entry for each task that has been queued for that map.

Some tasks, such as Profiling, may be listed even though you did not specifically queue them. Discovery executes these tasks as necessary to obtain correct results for a queued task.

You can sort the Activity List by any column (except Label) by clicking on the column header. The entries for each tab or map are sorted within that
3. Click the task’s expand box to display more information about that task.
   - The **Messages** tab displays a summary of the activities performed during that task execution. This is a less detailed list than what is contained in the Trace, Error, and Debug Log Files.
   - **Options** displays the option settings that were used for that particular execution.
Cancel a Task in the Activity Viewer

About this task

You can cancel any task while it is still queued, or during execution.

Note: Canceling the Discovery Engine task does not stop the database query. If the task involves a large or complex database query and you want to stop all work related to that task immediately, cancel the task and then terminate the database session itself. You can then continue working in Discovery Studio.
**Procedure**

1. Display the queued or running task in the Activity Viewer.

2. Click the icon. A confirmation dialog appears.
3. Click Yes to cancel the task. The status of the task changes to Canceled.
Delete a Task from the Activity Viewer

About this task

You can clear (delete) one or more completed or canceled tasks from the Activity Viewer. This can be useful when you have executed a number of tasks and want to clean up the Activity List.

**Note:** When you delete a task, its associated Trace, Error, and Debug Log Files are also deleted.

To delete a completed or canceled task:

**Procedure**

1. In the Activity Viewer, display the desired task(s).
2. Click the C (clear) to the right of the task(s) to be deleted. A confirmation dialog appears.
3. Click Yes. The task is cleared from the list, and the associated Trace, Error, and Debug Log Files are also deleted.
Status Icons

Discovery uses many Tab Status Icons to represent actions, status, or type. Hover the mouse over an icon to display a popup with information about it.

Status Messages

Status messages are displayed in several places in Discovery Studio. It is useful to keep the Error List and Output Pane open while working in Discovery Studio.

- In the Activity Viewer, you can display Trace, Error, and Debug Log Files associated with a specific task.
- The Activity Viewer also displays a less detailed summary of the action performed during a task's execution when you View Task Activity Summary.
- The Error List and Output Pane displays errors and warnings related to validation and other non-task related activities performed in the current Discovery Studio session.
- The Output Pane displays status messages related to work performed in the current Discovery Studio session.
Log Files

Log files are stored in the Discovery Server, Discovery Studio, and Discovery Engine installation directories. These locations are included here for reference, but information in these files is either already viewable in Discovery Studio, or is operational data relevant only to a support engineer.

- Log files in `<Discovery_Studio_install_dir>/bin` contain data about Discovery Studio actions. The contents of these files are shown in the Output Pane.
- Log files in `<Discovery_Engine_install_dir>/Log` contain data about Discovery Engine Service actions and Discovery Engine process actions. These can be viewed using the Discovery Engine Log Viewer.
- Log files in `<Discovery_Server_install_dir>/server/exeros/log` contain operational data about exceptions, connection errors, locked objects, and so on. These logs can be viewed using the Discovery Server Log Viewer.
- Log files in `<Discovery_Server_install_dir>/server/exeros/Logs/DELogs` contain data about task execution. This data is displayed in Discovery Server as Trace, Error, and Debug Log Files in the Activity Viewer. When you delete a task from the Activity Viewer, its associated trace, error, and debug files are deleted from this directory.

Trace, Error, and Debug Log Files

About this task

The Activity Viewer allows you to view the trace logs, error logs, and debug logs associated with each task.

- Trace logs contain detailed information about the Discovery Engine's actions during a task's processing.
- Error logs list any exceptions encountered during task processing.
- Debug logs contain information useful to a support engineer. They do not contain any information useful for working in a project. However, if you encounter problems executing a task, you may be asked to send these files to a support engineer.

Procedure

1. Open the Activity Viewer and select a particular task under Project Activities or Map Activities.
2. Click the arrow in the **View Log** button to display the available logs, and select the desired log from the list.
3. The selected log displays in a new window. If there is no data in the log, the window is empty.

```
[11:58:30 AM] Started Target Matches Analysis for Project demo, Source Data Source, NewSourceDataset and Target Data Source NewTargetDataset
[11:58:30 AM] Completed Target Matches Analysis for Project demo
[11:58:30 AM] Started Data Object Map Generation for Project demo, Source Data Source NewSourceDataset, and Target Data Source NewTargetDataset
[11:58:30 AM] DO Table Map DOMap0_DO_HQ_STORES_to_DO_W_DEPTS was generated.
[11:58:30 AM] DO Table Map DOMap1_DO_HQ_EMP_to_DO_W_EMPS was generated.
```

**Discovery Engine Log Viewer**

**About this task**

This log viewer displays Discovery Engine Service log files. These are not the Trace, Error, and Debug Log Files but instead are logs of internal Discovery Engine activities.
This tool is available from the Start menu of the machine where Discovery Engine Service is installed, at Start>Programs>IBM InfoSphere>Discovery>Discovery Engine>Discovery Engine Log Viewer.

When Logs are Created:

A new log is created at midnight every day. The Discovery Engine Service writes to the log every time it starts a Discovery Engine process, and records the Windows process ID (PID) of the process along with the outcome. The frequency of spawning Discovery Engine processes is configured in the Discovery Engine Configuration Utility, described in the IBM InfoSphere Discovery Installation Guide.

Log Filenames:

The filename contains the log file date in MM_DD_YYYY format, and the number of the Discovery Engine that the log is for. If only one Discovery Engine Service is installed on the machine, it is numbered DiscoveryEngine1. Additional Discovery Engine Services installed on the same machine have sequential numbers: DiscoveryEngine2, DiscoveryEngine3, and so on.

Viewing Logs:
By default, the log viewer displays the current or most recent log for DiscoveryEngine1. You can select a log for a different date using the dropdown menu on the log viewer. If more than one Discovery Engine Service is installed on the machine, a dropdown menu allows you to select a different service.

About this task

You cannot edit the log file.

Contents of the Log:

If the Discovery Engine process did not execute a task, the entries for that process are similar to the following:

6/15/2007 4:28:34 PM] This DiscoveryEngine is started with
pid:3528-31b77bf8c0a80274011c7a8c9d9bc663△[6/15/2007 4:29:40 PM] No work
scheduled for 1 minutes△[6/15/2007 4:29:40 PM] Finished processing run

If the process executed a task, the entries are similar to the following:

[6/15/2007 4:31:04 PM] This DiscoveryEngine is started with
pid:1796-31b9c4f9c0a80274011c7a8c7ab1f9fa△[6/15/2007 4:32:09 PM] Discovery
Engine ApplicationStartupPath is <C:\Program Files\IBM\InfoSphere\Discovery\Discovery Engine\Bin>△[6/15/2007 4:32:24 PM] Discovery engine
will log to the server every 180 secs OR 100 lines△[6/15/2007 4:32:09 PM] ProjectID:
31a9b5ac0a8027401ef3d1215167470△[6/15/2007 4:32:24 PM] ProjectID:
31b9d536c0a8027400f55ff219a871db△[6/15/2007 4:34:04 PM] Processed task 31b9d536c0a8027400f55ff219a871db with status Success

Searching the Log:

Type a string in the Search box and click Find to search the log for that string. Click Find again to locate the next instance of that string in the log.

Copying the Log Contents:

About this task

You can select text in the log file viewer and click Copy to copy it to the clipboard. To copy the entire contents, click at the beginning of the file, drag the slider bar to the end of the file, shift-click after the last entry, then click Copy.

Discovery Server Log Viewer

About this task

This log viewer displays Discovery Server log files. These are not the Trace, Error, and Debug Log Files, but instead are logs of internal Discovery Server activities.

This tool is available from the Start menu of the machine where Discovery Server is installed, at Start > Programs > IBM InfoSphere > Discovery > Discovery Server > Discovery Server Log Viewer.
When Logs are Created:

A new log is created when the previous log reaches a predetermined size. This depends on the server activity, so a Discovery Server log file can span up to several days.

The Discovery Server writes to the log when various Discovery Studio actions are performed, including opening some windows, changing options, and so on. The entries contain internal object numbers of projects, data sets, option sets, and so on.

Log Filenames:

The filename contains the log file number. The current log is always `exeros-server0.log.0`, with older files named `log.1`, `log.2`, and so on.

Viewing Logs:

By default, the log viewer displays the current or most recent Discovery Server log. You can view earlier or later logs by clicking Older or Newer.
About this task

You cannot edit the log file.

Contents of the Log:

Most Discovery Server log entries are similar to the following:

Jun 15, 2007 4:32:19 PM com.exeros.services.repository.RepositoryServiceImpl
getObjectHandles
FINE: Engine reading object handles HEAD_STORES/31b447ccc0a80274009e19fa998b0b09
for project 31a9b95ac0a8027401ef3d1215167470
Jun 15, 2007 4:32:19 PM com.exeros.services.repository.RepositoryServiceImpl
getObjectHandles
FINE: Engine reading object handles HQ_EMP_AL/31b849c6c0a8027402a39f17d71a0640 for
project 31a9b95ac0a8027401ef3d1215167470
Jun 15, 2007 4:32:20 PM com.exeros.services.repository.RepositoryServiceImpl
getObjectHandles
FINE: Engine reading object handles HQ_SALARIES/31b92b5ac0a80274012c760504303442
for project 31a9b95ac0a8027401ef3d1215167470
Jun 15, 2007 4:32:20 PM com.exeros.services.repository.RepositoryServiceImpl
getObjectHandles
FINE: Engine reading object handles TestProject1/31a9b95ac0a8027401ef3d1215167470
for project 31a9b95ac0a8027401ef3d1215167470

During task execution and other actions, entries may be similar to the following:

Jun 15, 2007 4:32:25 PM com.exeros.services.taskmanager.TaskManagerServiceImpl
isTaskCancelled
INFO: Checking whether the task is cancelled 31b9d536c0a8027400f55ff219a871db
Jun 15, 2007 4:32:25 PM com.exeros.services.taskmanager.TaskManagerServiceImpl
updateStatus
INFO: Updating status for task 31b9d536c0a8027400f55ff219a871db and action
31b9d536c0a8027400f55ff219a871db to Working
INFO: Updating status Column Analysis[Running]
Jun 15, 2007 4:32:25 PM com.exeros.services.taskmanager.TaskManagerServiceImpl
isTaskCancelled
INFO: Checking whether the task is cancelled 31b9d536c0a8027400f55ff219a871db
Jun 15, 2007 4:32:25 PM com.exeros.services.taskmanager.TaskManagerServiceImpl
addMessage
INFO: Adding message for task 31b9d536c0a8027400f55ff219a871db and action
31b9d536c0a8027400f55ff219a871db: Started Discover Datatypes.

Searching the Log:

Type a string in the Search box and click Find to search the log for that string.
Click Find again to locate the next instance of that string in the log.

Copying the Log Contents:

You can select text in the log file viewer and click Copy to copy it to the clipboard.
To copy the entire contents, click at the beginning of the file, drag the slider bar to
the end of the file, shift-click after the last entry, then click Copy.

How to Preview Data

Preview Criteria Screens

Most screens provide a popup dialog allowing you to view the data in the selected
table, or to view the results of applying a particular SQL expression to the source
and target tables.
In the Sample Sets and Column Analysis tabs, the **Preview Data** button displays the data. In the remaining screens with this feature, you select **Show Hits**, **Show Misses**, **Show Data**, **Show Duplicates**, or similar options.

A **Preview Criteria** screen appears that allows you the option of setting display criteria, including the number of rows to display, a where filter, and a sort order. When you have set the display criteria, or after you choose to skip it, the actual data is displayed.

The layout of the **Preview Criteria** screen is similar for all windows, but there are slight differences based on the type of data being displayed.

- **PF Keys Preview Criteria** and **Data Objects Preview Criteria** has a section for each related table, identified by name.
- All of the **Maps Preview Criteria** screens have two sections labeled **Source** and **Target** instead of specific table names.
- **Column Analysis Preview Criteria** and **Sample Set Preview Criteria** have only one section, displaying the name of the selected table.

**Select Preview Criteria**

**About this task**

The example below is of the **PF Keys Preview Criteria** screen, but you can use these instructions for all Preview Criteria screens.
To preview data:

**Procedure**
1. Click **Preview Criteria**. The *Preview Criteria* window appears, allowing you to customize the data and how it is displayed.
   - To preview all of the data in the table or expression, click **OK**.
   - To limit the number of rows displayed, create a where clause, or sort the output, continue with the following steps.
2. In the Limit result to field, select the number of rows to display or leave it at the default. The maximum number of rows that can be selected for display is 999999.

3. To filter the preview data in one or both sections using a where expression (optional):
   a. Begin typing in a Where field. A dropdown list of available tables appears.
   b. Double-click on the table name to select it.
   c. Type a period ( . ) to display a list of available columns for the selected table.
   d. Double-click on the column name to select it.
   e. Add any required punctuation by typing directly in the Where field.
   f. To delete some or all of a where expression, select it with the mouse and press the Delete key.
4. To display the data sorted in order by a column (optional):
   a. Click the + . An entry appears in the Order By list.

   b. Click the down arrow in the entry to display the list of columns in the sample set. Highlight the column to sort the previewed data by this column.
c. To preview the data sorted by several columns, create additional sort entries by clicking the + again. For example, create a City column sort and a Zip Code column sort to display the data sorted by city and by zip code within each city. Use the up and down arrow indicators to arrange the order of sort entries within the list.

5. **Show on Data Preview** determines whether this screen displays when Show Hits, Show Misses is clicked. By default, the box is checked and the Preview Criteria screen displays. Uncheck this box to keep this window from displaying. You can re-display this window by clicking Preview Criteria from a Data Preview screen.

6. Click **OK** to display the preview data.

   If you displayed the Preview Criteria screen by clicking Preview Criteria from a Data Preview screen, the **Refresh** button closes the Preview Criteria screen and updates the Data Preview screen. The **OK** button saves the changed criteria without updating the Data Preview screen.

**Results**

The data preview window opens. The format of the window depends on the type of data being displayed.

**Preview Criteria Icons**

**About this task**

The following icons are available on all Preview Criteria screens.

- **Add**
- **Delete**
- **Move Up, Move Down**
- **Show on Data Preview**: Uncheck this to prevent the Preview Criteria Screens from appearing the next time you open a Data Preview screen. To modify the preview criteria in a Data Preview screen, click the **Preview Criteria** button.

To have the Preview Criteria screen appear again, open a Data Preview screen and click **Preview Criteria**, then re-check **Show on Data Preview**.
Data Preview

All Data Preview screens allow you to sort, filter, focus and search the results. In some cases, you can edit expressions directly in the display and manipulate the data in other ways.

- Data Preview in Sample Sets and Column Analysis displays the contents of the selected table along with statistics.
- Data Preview in the **PF Keys** tab, **Data Objects** tab, and **Target Matches** tab applies the selected SQL expression to all data in the source and target tables and displays the hits, misses, duplicates, or all data, depending on the options available in each tab.
- Data Preview in the **Maps** tab applies the selected join, binding condition or transformation to selected columns and displays the hits, misses, duplicates, or all data, depending on the options available in each tab. In these screens you specify or drag specific columns into the table panes. You can also edit the expression in this preview and refresh the display to immediately view the results.

Common Elements in Data Preview Windows and Common Grid Icons in Data Preview Screens are described in the next section. Links to specific data preview windows are listed here.

- Logical Tables Data Preview
- Preview Sample Set Contents
- Column Analysis Data Preview
- Column Matches Show Hits, Show Misses
- Foreign Keys Data Preview – Show Hits, Show Misses
- PF Keys Foreign Keys Data Preview – Show Duplicates
- Data Objects Data Preview – Show Hits, Show Misses
- Source Mapping Data Preview
- Unified Column Analysis Data Preview
- Column Conflicts Data Preview
- Match and Merge Analysis Groups Data Preview
- Unified Schema PF Keys Data Preview
- Target Matches Data Preview – Show Hits
- Target Matches Data Preview – Show Misses
- Joins Data Preview: Show Hits and Misses
- Binding Conditions Data Preview – Show Data, Show Hits, Show Misses
- Transformations Data Preview (Show Data, Show Hits, Show Misses)

Common Elements in Data Preview Windows

About this task

Most Data Preview windows have the same layout and are similar in appearance. (Preview Sample Set Contents and Column Analysis Data Preview are different.) The Data Objects Data Preview – Show Hits, Show Misses screen is shown below as an example.

The dropdown menu by the **Columns** list indicates that the current display is for Show Hits. You can change this to **Show Misses** (or another preview type, depending on the screen) by clicking the dropdown arrow and selecting the new
The screen is divided into several panes:

- **Expression** displays the selected expression (binding condition, transformation, or other expression) that is applied to the data to obtain the results shown in the rest of the screen. In some screens, you can edit the expression and then click **Refresh Statistics** to update the display with the new results.

- **Parent** (or **Source** or **table_name**) displays data in one set of columns. The name of this pane varies depending on the screen. In some screens, you can add columns by dragging them from the Columns pane.

- **Child** (or **Target** or **table_name**) displays data from the other set of columns. The name of this pane varies depending on the screen. In some screens, you can add columns by dragging them from the Columns pane.

- **Columns** displays the columns involved in the expression, grouped by table. Depending on the screen, these may include source columns, target columns, or both.

To remove a column from a table pane, grab it by the header and drag it downwards. Toward the bottom of the pane, the cursor turns into an X. Release the mouse and the column is removed from the source table pane. (It is not deleted from the source table or the discovered results.)
Common Grid Icons in Data Preview Screens
The following grid icons are common to most Data Preview screens.

- Advanced Search
- Search
- Auto Filter Row
- Column Chooser
- Export Grid Content
- Focus

Common Buttons in Data Preview Screens
The following buttons are common to most Data Preview screens.

Preview Criteria:
Click this to redisplay the Preview Criteria Screens.

Refresh Query:
This updates the contents of the Query pane with the results of the modified expression. See Transformations Data Preview (Show Data, Show Hits, Show Misses).

Refresh Statistics:
This updates the contents of the Parent (or Target) pane and the Child (or Source) pane with the results of the modified expression. See Common Elements in Data Preview Windows.

Show Hits/Show Misses/Show Data:
This changes the contents of the Data Preview screen. Selecting a different view here is the same as closing the Data Preview screen and clicking Show Hits, Show Misses, Show Data. The current preview criteria is applied to the new view.

Apply:
Saves the changes to the current expression. This is useful after refining an expression to improve the results.

**Save as New:**

Saves the current expression as a new entry in the grid. If the same expression already exists, this is added as a new entry (the existing expression is not overwritten).

## The Expression Editor

The Expression Editor is an editing field for creating and editing SQL expressions. In some windows, such as Transformations, it is a permanent field on the screen; in other windows, like PF Keys, it appears when you click on an editable SQL field.

### Invalid Expressions

Discovery performs validation on expressions as you enter or modify them. Invalid expressions are indicated by an X in the grid. The statistics for invalid expressions may be stale.

You can save and close a project containing an invalid primary expression. Statistics for invalid expressions also cannot be refreshed.

If the primary join condition, primary binding condition, or primary where clause is invalid, the entire map is considered invalid and all statistics in the map are stale.

## Using the Expression Editor

Expressions are validated during entry.

### About this task

A red X in a circle appears if the expression is invalid. Correct the expression before continuing.

### Procedure

1. Display a window or screen with an editable SQL field. The Expression field in the PF Keys Window is shown here.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Row Hit Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>HQ_EMP.EMPLOYEE_ID = HQ_SALARIES.YEARS_AT_WORK</td>
<td>29% (9/31)</td>
</tr>
<tr>
<td>HQ_EMP.EMPLOYEE_ID = HQ_SALARIES.EMPLOYEE_ID</td>
<td>90% (28/31)</td>
</tr>
</tbody>
</table>

2. Left-click in the expression. The Expression Editor appears.
3. Edit the expression in the Expression Editor.
   - Highlight, type, or delete the expression to modify it.
   - To select a table from a list, press Ctrl-Space or start typing in the field. A dropdown list of valid tables appears. Use the down-arrow to select from the list, then press Enter to add it to the expression.

4. To select a column from a list, type a period (.) immediately after the table. A dropdown list of valid column names appears. Select from the list using the down-arrow, then press Enter to add the column name to the table.

4. Click outside of the Expression Editor to accept the changes and close the Expression Editor. (Click the red X to close without saving changes.)

5. Click Refresh Statistics to make Discovery re-run the statistics using the new expression.

Results

You can also use + to manually compose a whole SQL expression, which you can then validate and add to the list.
Menus, Screens and Commands

This section describes elements in the Discovery Studio interface that are not specific to a particular tab, but that appear on many or all tabs, popups, and other screens.

Main Menus

Discovery Studio contains several main menus that are always available, and other menus that appear only when certain tabs are selected. Some menu items or options are available only in TD or SDD projects.

• Project Menu
• View Menu
• Tools Menu
• Data Rules Menu
• Map Menu
• Help Menu

Project Menu

The Project menu is available in every tab. Certain project-level options are clickable only when a project is open, such as Validate, Export, and Save.

If a menu option is greyed out, it is not available in that type of project, or cannot be selected in the project's current state.

Validate

Validates all data in the project. This is different than the Validate Step button, which checks for syntax and other errors only in the current step.

Import

Import Full Project allows you to import a previously exported Discovery project into a new, empty project. This option is available at all times.

Import TD Project Objects allows you to import selected objects, such as maps or data rules, into existing projects.

“Importing Business Glossary XML” on page 281 allows you to import previously exported Business Glossary XML files.

“Importing from Guardium” on page 194 allows you to import previously exported selected objects into existing projects.

Export

This feature allows you to export part or all of a project in various formats.

• Full Project: Export Full Project to create a backup or for importing into another project.
• Project Objects: Export Project Objects to create a backup or for importing into another project. (TD projects only)
• Export for Optim: Generate Optim Export
• Export to Guardium: “Exporting to Guardium” on page 593
• Export for FastTrack: “Export for InfoSphere FastTrack” on page 599
• Export for Metadata Server: "Export for Metadata Server" on page 601
• Business Glossary XML: (SDD projects only) "Export Business Glossary XML"
on page 601
• Discovery XML: Generate Discovery XML
• CWM XML: Export CWM XML

Save

Validate and saves the currently open project. Discovery also saves the project at various times while you are working in the project, such as before executing a task.

Exit Studio

Exits Discovery Studio. You are prompted to save an open project if you have made changes since the last save.

You do not need to exit Discovery Studio to Close a Project or Connect to a Different Discovery Server.

View Menu

The View menu is available in all tabs, but the options are only clickable when a project is open.

These options provide an alternative way of opening the Discovery Studio tabs. Click on an option to open the tab, or use the keyboard shortcuts shown.

If a menu option is greyed out, it is not available in that type of project, or cannot be selected in the project's current state.

TD View menu options:
• Home - Ctrl-1
• Data Sets - Ctrl-2
• Column Analysis - Ctrl-3
• PF Keys - Ctrl-4
• Data Objects - Ctrl-5
• Target Matches - Ctrl-6
• Maps
• Column Classification View - Ctrl-7

SDD View menu options:
• Home - Ctrl-1
• Data Sets - Ctrl-2
• Column Analysis - Ctrl-3
• PF Keys - Ctrl-4
• Data Objects - Ctrl-5
• Overlaps - Ctrl-6
• Unified Schema - Ctrl-7
• Unified Schema PF Keys - Ctrl-8
• Column Classification View - Ctrl-9
Open Maps
The Open Maps menu option is displayed only after maps have been added to a project, and is available only when a project is open. Each map in the project is listed. Click a map name to open the most recent Maps tab for that map.

Tools Menu
The Tools menu is available in every tab. The Reports options are clickable only when a project is open.

If a menu option is greyed out, it is not available in that type of project, or cannot be selected in the project’s current state.

Tools>Options> General
These options apply to all projects created in Discovery Studio.

Maximum lines in text file import wizard:

Determines the maximum number of lines displayed during [Import Text Files] when you click Text Preview in the import wizard.

Handling long database table numbers when DB2 is staging database:

DB2 allows a maximum numeric precision of 31 (the total number of places on both sides of the decimal point). When importing columns from other databases that allow larger precision, DB2 may not stage those numbers correctly. For example, Oracle and Sybase allow a maximum of 38 places.

When a column with a precision greater than 31 is imported into a project that uses a DB2 staging data source, you have two options:
• The number can be truncated to a precision of 31.
• The entire column can be treated as a VARCHAR data type. Discovery cannot perform arithmetic discovery on VARCHAR columns. However, the column may be staged correctly as a numeric datatype during the Column Analysis step if the discovered precision is 31 or less.

User Information:

This is the user name shown in the Notes entries for all projects in this Discovery Studio instance. By default, it is the Windows logon name of the account that installed this Discovery Studio on the machine.

Any new name you enter here will be used for all future Notes entries in this Discovery Studio instance. The user name in existing Notes will not be changed.

Other Options:

Allows you to suppress the automatic reminder to save the SDD project after making changes in the Value Overlap Details.

Tools>Options> Default Staging Data Source

These options apply to all projects created in Discovery Studio.

Creating a default staging data source can save time when several projects will use the same staging data source. Instead of manually entering the staging data source
information for each new project, you can select the default staging data source. Both SDD and TD projects can use the same default staging data source.

Database Types and Versions for Staging Data Source and Repository:

The staging data source and repository for each project must be of the same database type and version. If the default staging data source does not match the current Discovery Server’s repository, you must manually define a different staging data source when creating a project.

Procedure
1. From the Tools menu, select Options.

2. In the Options dialog, click Default Staging Data Source.
3. Fill out the fields in the Default Staging Data Source dialog as described in Step 5 of the Create a Project procedure.
4. Click OK to close the Default Staging Data Source dialog.
Results

The default staging data source is now configured and ready to use in projects.

Change the Default Staging Data Source:

The default staging data source can be changed at any time, even if you have already assigned it as a project's staging data source.

About this task

Changes to the default staging data source only affect new projects. Existing projects that specified the old default staging data source continue to use the previous schema. The only way to change an existing project's staging data source is to Edit a Project's Properties.

To change the default staging data source, open the Tools>Options menu, select Default Staging Data Source, and edit the properties as described in the section above.

Tools>Options> Import Processing

These options apply to all projects created in Discovery Studio.

Allows you to specify SQL commands for Discovery to perform on columns of certain data types when adding a table or text file to any data set. You can specify different SQL commands for columns of string data type, numeric data type, and date data type.

When you delete the SQL command, Discovery no longer performs that processing on newly-imported tables or files. However, tables and text files that have already been imported and processed with those commands are not changed.

You can Define a Logical Table that has custom import processing commands by using the Add a new logical table with processing option in the Data Sets tab.

Tools>Reports

This option generates reports in various formats containing information about the open project.

- Project Report
- Table Report
- Column Report

Tools> Validation Jobs

(TD projects only) This option creates jobs and executables for use with Exception Manager. See Create Validation Jobs.

Tools> Clear All User Settings

Grid filters, grid layouts, and diagram layouts are saved in the current user’s Windows profile. On rare occasions, these settings become corrupted and need to be cleared by closing the project and selecting this option. This has no effect on the project itself, but only returns the grid and diagram displays to their default state.
**Tools>Unified Schema Table Copy**

(Unified Schema tab in SDD projects only) This option makes a copy of the selected target table schema and adds it to the Target Table Schema List.

**Data Rules Menu**

This menu is displayed only when the Data Sets tab is selected. It is available only in TD projects.

**View**

Select this item to display the Data Rules Window.

**Map Menu**

This menu is displayed when any Maps tab is selected (Summary, Joins, Bindings, etc.) It is available only in TD projects.

**Copy**

Select a map in The Maps List and then select this option to create a copy of the map. The copy is added to the same target table group.

**Advanced - Generate BC Report**

This option allows you to Generate BC Report.

**Help Menu**

**Online Documentation**


**About Discovery Studio**

Displays the build version of Discovery Studio.

**Common Window Elements**

Most windows in Discovery Studio have a number of common features, commands, and displays. These are described below.

**Note:** This section use a TD project screenshot. The elements are the same for SDD projects.
The following elements are common to all Discovery Studio displays.

**Main Menus**
The menus along the top of the screen contain project-level and application-level options.

**Current Discovery Server Host**
The name of the Discovery Server host that this Discovery Studio instance is connected to. LOCALHOST indicates that Discovery Studio is running on the same machine as Discovery Server.

**Name of Open Project**
The project that is currently open. If no project is open, this reads No Project Loaded.

**Activity Viewer**
Displays a screen that allows you to monitor and review tasks.

**Validate Step**
Checks for syntax errors in expressions and performs other validation for the step. Displays results in the Output Pane.
**Approve Step**

(Unavailable while a task is running) Click this to mark this step as approved and change the Tab Status Icons to a checkmark. This indicates you have completed all discovery and review tasks in this step. After a step is approved, this button changes to **Un-approve Step**.

If you perform any actions in this step after marking it as approved, Discovery automatically removes the Approved status. You can also manually remove the status by clicking **Un-approve Step**.

**Re-Run Step**

Repeats the discovery actions for the current step. See [Move from One Window to the Next](#).

**Run Next Steps**

Configures and executes the discovery actions for the next step. See [Move from One Window to the Next](#).

**Error List and Output Pane**

Contains status and error messages.

**Resize the Discovery Studio window**

Drag any outside border of the Discovery Studio window to increase or decrease the size of the window. Panes and displays in the window adjust to accommodate the new size. If the window is very narrow or short, some buttons or icons may be covered up, or you may need to scroll to see the full content of a pane.

**Window Panes**

Windows are divided into several sections, or panes. Each type of pane has the same characteristics and many of the same features.

Each window contains two or more of the following panes:

- **Error List and Output Pane** displays error and warning messages related to executing tasks. All windows have an Error List.
- **Output Pane** contain status and output messages related to other Discovery Studio actions, such as validating steps. All windows have an Output pane.
- **List Panes** display a list of logical tables, maps, or other objects in the window. All windows have at least one list pane.
- **Grid Panes** display statistical and other information about objects in the list pane. Most windows have grid panes.
- **Diagram Panes** display a graphic representation of objects in the list pane.

A window can have two or more of the same type of pane. For example, the TD project Column Analysis window below contains three panes: a list pane, a primary grid pane, and a secondary grid pane. The Messages pane is collapsed.
As another example, the SDD project PF Keys window below contains a list pane, a diagram pane, and a grid pane.
Because Discovery Studio contains numerous panes showing a great deal of information, a number of features are provided that let you modify the size and location of panes.

**Autohide Pane**

**About this task**

You can use the `autohide` function to collapse the Maps List and the Error List and Output Pane. This gives more room onscreen, and you can temporarily redisplay the pane by mousing over the tab.

**Procedure**

1. Click on the vertical pushpin icon.
The pushpin becomes horizontal and a new tab appears on the left border of the pane. Also, the title bar becomes light blue to indicate that autohide is active.

2. Move the cursor off of the pane. The pane collapses to the nearest border of the Discovery Studio window.

3. To temporarily redisplay the pane, move the cursor over the tab. The pane expands, and then collapses again when you move the cursor away.

4. To turn off autohide, expand the pane and then click the horizontal pushpin. The pushpin becomes vertical again, and the pane remains displayed when you move the cursor away from it.
The Expand/Collapse Button

About this task

A Discovery Studio window usually contains several panes. Between some panes is an Expand/Collapse button, which allows you to expand one pane and collapse the other to increase your available workspace.
The arrow in the Expand/Collapse button points in the direction of the collapsing pane. In the screenshot above, it points downward — when you click this button, the upper pane expands and the lower pane collapses.

Clicking the button again restores the original layout.

**Dragging Boundaries**

**About this task**

You can manually resize panes by dragging their boundaries.
Error List and Output Pane

This section describes the error list and the output pane.

The following icons are used in the Error List and Output List:

- Error List Display
- Output Pane Display
- Output Pane Clear
- Word Wrap On/Off

Error List

The Error List displays error and warning messages related to validating expressions. It also contains other errors and warnings encountered by Discovery Studio that are not related to executing tasks.

- Errors are problems that must be fixed before processing can continue.
- Warnings do not affect processing, but you should be aware of these issues and consider fixing them before continuing.

Displaying Errors and Warnings:

You can display only errors or only warnings by unchecking the Errors or Warnings icons.
### About this task

Errors and Warnings icons checked

Click on the **Errors** or **Warnings** icon to check or uncheck it.

Errors icon unchecked; only Warnings are displayed

**Clearing Errors and Warnings:**

You do not manually clear (delete) errors and warnings from the Error List. The errors and warnings for each tab remain in that tab’s Error List pane until you execute another task in that tab. At that time, the previous errors and warnings are replaced with any new ones. If there are no new errors or warnings, the Error List becomes empty.

For example, assume you are in the **PF Keys** tab and there are no errors or warnings. You execute **Run Next Steps**, and an error occurs. The error is shown in the Error List of the **Data Objects** tab. If you click back on the **PF Keys** tab, the Error List becomes empty. But when you click on the **Data Objects** tab again, the Data Objects task error reappears in the Error List. That error remains in the list until you re-run the Data Objects step, or until you return to the **PF Keys** tab and execute **Run Next Steps** again.

### Output Pane

The Output pane shows messages related to general Discovery Studio operation. Selecting an option does not filter the existing messages, but determines the kinds of messages that are logged from then on.

- **All** — information, debug, warning, and Discovery Studio error messages
- **Normal** — warnings only
- **Errors Only** — Discovery Studio errors only
The Output pane contains all output messages for the project in the current Discovery Studio session. Each new output message is added to the end of the list. You can scroll back through the Output pane to view all messages for that project in the current Discovery Studio session.

**Clearing Output:**

When you close the project, the Output pane automatically clears.

**List Panes**

List panes display the objects in the window by name, rather than graphically as in the Diagram Panes. The examples below are from both TD projects and SDD projects.
All list panes allow you to Filter Items in a List Pane. In most list panes, you can also add and delete items. Instructions for adding and deleting are in the specific tab's description:

- Column Analysis Tables Pane
- PF Keys List
- Data Objects List
Filter Items in a List Pane

You can filter the tables in a list pane by string. Only one filter can be applied at a time.

Apply a Filter:

To apply a filter to a list pane:

Procedure
1. Enter the string into the filter field.
2. Click the Apply Filter button. The list is filtered to contain only the entries containing that string.

Results

In the example below, Discovery is filtering the PF Keys list for the string EMP. The filtered list shows only entries with EMP in the name. In all list panes, the main object (such as the main PF Key entry, data object, Target Match, etc.) of a filtered item is shown, whether or not it also meets the filter criteria.
The filters you create remain available in the filter dropdown menu for the duration of this Discovery Studio session.

Clear a Filter:

To remove the current filter and display all tables or objects, click the filter dropdown menu and select <Clear Filter>. This option is only available when a filter is currently applied.

About this task

Diagram Panes

The PF Keys, Data Objects, and Target Matches windows use diagrams to visually represent relationships and objects.
All diagram panes share the following characteristics:

- Colors of Source and Target Data Sets
- Selecting Tables and Connections
- Columns in the Diagrams
- Rearranging the Diagram

**Colors of Source and Target Data Sets**

In SDD projects, all icons and lines are blue.

In TD projects, icons and lines for tables and connections in the source dataset are blue, and icons and lines in the target dataset are orange.

**Selecting Tables and Connections**

The objects in the diagram are associated with the objects in the list pane.

**About this task**

When you click a table name, relationship, or object in the list pane, the associated image is highlighted in the diagram. Likewise, when you select a table or connection in the diagram, the associated entry is highlighted in the list pane.

When an item in the diagram is selected, information about it (if available) is displayed in the grid below the diagram. In the example below, a PF Key connection is selected in the diagram. The corresponding PF Key is highlighted in the Connected Tables list, and all discovered expressions for the PF Key are shown in the grid below the diagram.
Columns in the Diagrams
In the diagrams, the columns in the tables are view-only. You cannot select a column by clicking on it in the diagram, and any actions you perform on a table in the diagram apply to the entire table, not a particular column.

In some diagrams, you can create table connections by drawing lines between tables in the grid. Drag and drop the line anywhere within the white space of the table. After creating the connection between tables in the diagram, identify the specific columns in the connection using the Grid Panes.

Columns with a relationship to a column in another table are shown with a [Primary Key] icon.

Rearranging the Diagram
The PF Keys, Data Objects, and Target Matches windows offer several features for viewing and rearranging the diagram.

Zoom:
You can adjust the view from 40% of normal size (zoomed out) to 160% of normal size (close up).

The Zoom to Fit option automatically resizes the diagram to display all current objects at the largest size possible.
Changing the Table Size:

By default, three columns are shown in each table. You can display more of the columns in the table by dragging the side and bottom borders of a table.

About this task

Moving Tables:

To move a table, click in the table's title bar and drag it to the new location. All connection lines are resized or rearranged to best fit in the new location.

Moving Connections:
You cannot manually change the location where a connector touches a table border. However, moving a table may change the connection path or location.

**Printing and saving diagrams**
You can print diagrams to a printer or save diagrams as bitmap images.

**About this task**
Before you print the diagram, arrange the layout to display the desired content.

**Procedure**
1. Click **Print** at the top of the diagram.
2. Select the appropriate option:
   - **Print** Allows you to select a hard copy (paper) printer or a print driver, such as a PDF print driver. Large diagrams are printed as multiple pages.
   - **Print Preview** Displays the diagram exactly as it will appear when you print. You can change the page orientation and scale, and print directly from the preview.
   - **Save as Image** Saves the diagram as a bitmap graphic file. Large diagrams are saved as a single file.

**Grid Panes**
Some of the data that Discovery Studio discovers is displayed on screen in a grid format. Grid panes display data about the table, relationship, or other object that is selected in the list pane or in the diagram pane.

Most grids in Discovery Studio use the same format and provide many of the same features for viewing and manipulating the data. These are described in the following sections.

**Column Chooser**
**About this task**
In many tables, additional columns of statistics are calculated but are not shown by default. The **Column Chooser** displays a list of items that can be shown or hidden in a grid.
You can change the sort order in the Column Chooser by clicking the Band and Column headers. To return the columns to the original order in the Column Chooser, sort by the # column.

Columns and Bands:

Items are in two categories: Columns and Bands. Columns and bands that cannot be hidden are not listed in the Column Chooser.
Export Rows and Copy Rows:

Showing or hiding columns and bands from the display does not affect the data or the results of any discovery tasks. However, [Export Grid Content] or [Copy Rows] only includes the columns and bands that are currently displayed in the grid.

Refreshing Statistics in Hidden Rows:

When discovery is run or re-run on a step, all statistics are refreshed, even statistics in hidden columns. When the column is redisplayed, the updated statistics are shown.

Hiding and Showing Columns and Bands:

About this task

To hide a column or band in the grid, uncheck its checkbox in the Column Chooser. To show it again, check its checkbox.

If you check a column to unhide it but it does not display in the grid, make sure that its band is not hidden. Columns in a hidden band cannot be displayed. However, a band can be displayed even though all of its columns are hidden.

You can select multiple rows in the Column Chooser by clicking and dragging, or use CTRL-click to select non-adjacent rows. While multiple columns are selected, rt-click to display a dropdown menu allowing you to check all of the selected columns, uncheck all of them, or toggle the columns (check the unchecked ones, and uncheck the checked ones).
To hide an entire band and its columns, click the Bands tab and uncheck the band. In the following example, unchecking the Metadata band and closing the Column Chooser hides the Metadata band and all of its columns. Only the Statistics and Notes bands are shown in the grid.

Note that when you hide a band, all of its columns are automatically hidden.
Notes
Some of the screens, dialogs, and grids have a Notes field where you can enter a description, comment, or other notes about the object.

- Enter Notes
- Modify and Delete Notes
- Change the User Name
- Export Notes

A Notes field in a screen or popup is usually near the top, as shown below.

Enter Notes:
Click on the Notes icon or the dropdown arrow to display the Notes entry box. This box is resizable.

About this task
Discovery Studio automatically prefaces each entry with the name of the currently logged in Windows user, the date, and the time. You can change this information; for example, if you are logged in under a different username, you can change the name to your own. After entering the note, click OK to close the entry box.
The Notes icon turns dark when it contains an entry, so you can easily identify which Notes fields are populated.

Modify and Delete Notes:

Each additional entry in the note is placed above the previous entry, so the most recent one is at the top. In subsequent entries, you can modify or delete the earlier entries. If all entries are deleted, the Notes icon becomes grey again.

Change the User Name:

By default, the user name is the Windows logon name of the person currently logged on to the machine.

The user name can be changed in the Tools>Options menu. The name entered here is used for all Notes entries in this instance of Discovery Studio on this host, regardless of who is actually logged on to the host or who is working in the project.
Export Notes:

Notes are included in all reports and exports that include the object. If you export a project or map containing notes and then reimport it into Discovery Studio, notes are also imported.

**Auto Filter Row**

**About this task**

When you click the Auto Filter Row icon, a new filter row appears directly under the column headers. Type your filter criteria into each column’s filter row as needed. The row is filtered as you type.

Click the icon again to remove the filter row.
In some grids, a filter icon appears after you start to type in the filter row. Click the icon to display a dropdown menu.

![Show All Matches](image)

The menu displays the following filter options:
- **Custom**: Allows you to specify one or two mathematical operators and one or two additional criteria that you can apply to the current column
- **Blanks**: Displays only blank rows
- **Non-blanks**: Display only rows that are not blank

Depending on the characteristics of the column, each unique value in the column may also be listed.

To clear a filter, delete the string in the filter row or right-click in the column header and select **Clear Filter**. Also, closing the filter bar clears all filters.

**Export Rows**

**About this task**

The **Export Grid Content** icon allows you to export the contents of a grid to a file in a variety of formats: comma-delimited, tab-delimited, Excel, HTML, or XML. You can also **Copy Rows**.

You cannot select individual columns for export; all columns in the selected row(s) are exported.

A supported version of Excel must be installed on the Discovery Studio host host, and must be configured to print. See the *IBM InfoSphere Discovery Installation Guide* for the supported versions.

To export all rows in a grid, select **Export All Rows**. Cells containing commas (,) are enclosed with double quote marks when exported.
You can perform a partial export of the grid in the Data Preview and Show Hits/Misses windows. Select a range of rows by clicking and dragging, or select non-adjacent rows using CTRL-click, and then select Export Selected Rows.

The full contents of the rows are exported exactly as displayed onscreen, including sort order, column order, and filters, even if you need to scroll up or down to see them. However, any columns or bands hidden using the Column Chooser are not included in the export.

**Primary Expression**

**About this task**

The Discovery grids display all of the discovered results which pass the thresholds for that task. In most of these grids, Discovery uses algorithms that involve calculations not displayed onscreen to select one result as the primary result. You cannot select more than one primary, and if only one entry exists, it is automatically the primary.

A selected button indicates the primary entry in a grid. The Transformations Grid has a slightly different layout.)

All other entries are deselected. Only one can be selected at a time.

The following example shows the Binding Conditions grid, with the last entry marked as the primary.

<table>
<thead>
<tr>
<th>Binding Condition</th>
<th>..</th>
<th>/</th>
<th>..</th>
<th>Not...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
<td><img src="image.png" alt="Image" /></td>
</tr>
<tr>
<td>+ (HQ_EMP.DOH = W_EMPS.BEGIN_DATE) AND (HQ_EMP.NAME1 = W_EMPS...</td>
<td>On...</td>
<td><img src="image.png" alt="Image" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ (HQ_EMP.NAME2 = W_EMPS.EMPLOYEE_LAST_NAME)</td>
<td>Ma...</td>
<td><img src="image.png" alt="Image" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ (HQ_EMP.DOB = W_EMPS.DATE_OF_BIRTH) AND (HQ_EMP.NAME2 = W_E...</td>
<td>On...</td>
<td><img src="image.png" alt="Image" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ (HQ_EMP.DOB = W_EMPS.DATE_OF_BIRTH) AND (HQ_EMP.NAME1 = W_E...</td>
<td>On...</td>
<td><img src="image.png" alt="Image" /></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For example, the “best” PF Key between two particular tables out of all of the alternatives is selected as the primary PF Key for those two tables, the “best” transformation between a target column and all matching source columns is selected as the primary transformation for that target column, and so on. It is very important to review the primaries in each step because your criteria may differ slightly from Discovery’s criteria. Also, you may be able to modify the expression to improve the statistics or create a new one with better statistics.

A few grids do not have a primary because the concept does not apply in that context. For example, there is no primary Target Match because target matches describe all available column relationships and are the basis for maps discovery.

When a grid uses primaries, subsequent discovery is performed based on the selected primary entry. You can change the primary by clicking on another entry’s button. If you modify a primary or select a new one, re-run discovery for all subsequent steps.
**Right-click Menu**

Right-clicking on a column header displays a dropdown menu listing the functions available for that column, such as sorting the table, grouping the rows, and resizing the columns for optimal viewing. If there is a filter applied to the table, this menu also includes a Clear Filter command. Functions that are not available for a particular column or table are greyed out or not shown.

**Copy Rows**

**About this task**

To copy some or all rows in a grid to the Windows clipboard, select a range of rows by clicking and dragging, or select non-adjacent rows using CTRL-click. You can also click any cell in the grid and press CTRL-A to select the entire grid. Then press CTRL-C to copy the selected data to the Windows clipboard.

**Note:** You cannot select individual columns for export; all columns in the selected row(s) are exported.

The data can be pasted into any appropriate application. The full contents of the rows are copied exactly as displayed onscreen, including sort order, column order, and filters, even if you need to scroll up or down to see them. However, any columns or bands hidden using the Column Chooser are not included in the copied data.

Also see [Export Rows](#)
Sort Rows by Column

Some grids are sorted by default by logical table name or physical table name. You can always sort the rows in a grid according to the data in any column, in ascending or descending order. You can also click on the column header to toggle between ascending and descending sort.

Note: In certain columns, some numeric data is sorted as a string.

Sort Rows in Ascending Order:

About this task

Left-click in the header of the column you want to sort the rows by. Some grids have columns that are grouped by bands; the band is not used when changing the sort order of the column.

An up arrow indicator appears, indicating the rows in the grid are now sorted according to the data in the column. The first click always sorts in ascending order, as shown in the following example.

Sort Rows in Descending Order:

About this task

Click the column header again to sort the rows in the opposite direction. The down arrow indicator indicates the new order. In the following example, the rows are sorted in descending order according to the data in the column.

Clear Sorting:
About this task

To restore the column to its original pre-sort order (unsort), right-click in the header and select Clear Sorting. Or, press the CTRL key and click on the arrow indicator. The arrow indicator disappears.

**Rearrange Column Order**

About this task

You can rearrange the order of columns in most grids by dragging a column header to a new position in the grid. When a highlight appears in the header of the destination, release the mouse. The column moves to the new position.

---

**Group By Columns**

Most grids allow you to reorganize the displayed data by one or more columns. This is a useful way of organizing large amounts of data for easier review.

**Group By a Column:**

Right-click in a column header. If **Group By Column** appears in the dropdown menu, you can use this feature.
About this task

The example above shows three values appearing several times in the Right Table column: HQ_EMP, HQ_SALARIES, and HQ_STORES. Right-click in the Right Table header and select Group By This Column to organize the rows in the grid into three groups, as shown below. New row separators indicate how the rows are grouped.

Group By Additional Columns:

You can set additional Group By criteria to further organize the rows. For example, right-clicking in the Right Column column and selecting Group By This Column sorts each existing group of rows according to the values in the Right Column.
Groups that only have one item in this column are unaffected, but groups with more than one value in this column are now broken into subgroups, based on these values.

About this task

You can organize by as many sub-groups as you like.

Sort Within Groups:
About this task

Click on any column header to sort rows by column. For all groups containing more than one value in that column, the values within the group are sorted in the new order.

The order of the groups and subgroups themselves are not changed.

Expand and Collapse Groups:
About this task

Each row separator has a +/- indicator in front of it. This allows you to collapse and expand the group in order to minimize unnecessary information on screen.
To expand or collapse all groups at once, right-click in a column header and select **Full Expand** or **Full Collapse**.

**Clear the Groups:**

To remove a particular grouping, right-click in the column header again and select **Ungroup**. All other groups remain active until you select **Ungroup** for them.

**Group By Box:**

As an alternative to right-clicking in the column header and selecting **Group By This Column**, you can right-click in any column header and select **Group By Box**.

**About this task**

This displays a working area above the headers where you can drag and drop column headers to create groups.

**Note:** When the working area is displayed and you right-click on a column header to group by that column, the column header is automatically added to the working...
For example, to group the grid by the **Right Table** column, click on the **Right Table** column header and drag it up into the working area, then release.

The column header is shown in the working area, and the grid is grouped by that column.
Drag additional headers into the working area to create subgroups. Once two or more headers are in the working area, you can change the groups and subgroups in the grid by rearranging the headers in the working area.
To clear a group, drag the header from the working area back into any part of the header bar. The grid is no longer grouped by that column.

To clear all groups at once from the working area, right-click in the working area and select Close Grouping. All headers are removed from the working area and the grid is ungrouped.
Change Column Width
About this task

In some Discovery Studio grids, column values are so long that the data is truncated in the on screen view.

You can place the mouse cursor over the truncated column value or heading to display the full text in a popup box, or you can manually resize the columns by dragging the borders in the column headers. You may need to scroll left or right to view the entire table.

You can also automatically resize columns by right-clicking on the header and selecting one of the following options from the popup menu:

- **Best Fit** resizes the current column to the width of the longest value in the column.
- **Best Fit (all columns)** resizes the all columns for the best view within the existing width of the table.
- **Size to Data (all columns)** resizes all columns in the table to the width of the longest value in each column, excluding the column header.

About projects

The first step in using Discovery is creating a project in Discovery Studio. All discovery, processing, analysis, and other tasks for a particular set of files are performed in a project.

A project contains all of the information about a specific group of tables and files. The information stored in a project includes the following:

- connections to the source and target tables
- locations of text files
- connection to the staging data source
• the location of all source files
• the history and status of all processing tasks
• all discovery results
• maps, for Transformation Discovery Projects
• overlaps and unified schemas, for Source Data Discovery Projects

Each project is completely separate from the others. Because Discovery never modifies the original source tables and files, the same tables and files can be used in multiple projects at the same time.

Types of projects
You can create two types of projects: Transformation Discovery (TD) projects and Source Data Discovery (SDD) projects.

Both types of projects use the same preliminary discovery steps to identify the contents of each data set and determine basic relationships between and within the data. You select the project type when you Create a Project. You cannot change a project’s type after you create it. Transformation Discovery Projects and Source Data Discovery Projects are listed separately in The Home Tab.

Where projects are stored
This section provides general information about SDD and TD projects.

The project file is stored as part of a schema in the current Discovery Server’s Repository Database. When you start Discovery Studio, the Discovery Studio Home tab displays all projects in that Discovery Server’s repository.

If you have several Discovery Suites at your site, be aware of which repository your Discovery Server is connected to when you create a project in Discovery Studio. (The current Discovery Server host is shown in the Discovery Studio title bar.) You cannot access projects in one repository when your Discovery Server is connected to a different repository.

Note: The repository database is different from the Staging Data Source. The staging data source is a work area where Discovery creates and manipulates copies of the user data files (both tables and text files) in a project. The staging data source and the repository can share the same physical database, and they must be of the same database type and version (such as Oracle 10g), but they should not be in the same schema.

About staging data sources
Every project uses a staging data source. This is a database used by Discovery as a work area, where copies of tables and text files in a project are stored and manipulated.

Default staging data source
To save time when several projects will use the same staging data source, use Tools>Options> Default Staging Data Source to create a default staging data source. You can also Change the Project’s Staging Data Source if needed.
The staging data source and repository for each project must be of the same database type and version. If the default staging data source does not match the current Discovery Server’s repository, you must manually define a different staging data source when creating a project.

**Managing a staging data source**

Discovery cleans up the project’s staging data source when a project is deleted. You normally do not need to manually delete objects from a project’s staging data source.

However, it is possible to manually delete objects from a staging data source. This can improve database performance or remove staging tables for old or rarely used projects. Discovery will recreate the deleted staging objects when they are needed, but it can take a long time to recreate large objects.

For information, see [Clean Up the Project’s Staging Data Source](#)

**About scan formats**

Scan formats are the rules that Discovery uses to convert strings to date-time or numeric formats. The default scan formats in each project are date, time, and number formats that are a good representation of local numeric and date-time formats for the region.

During data type discovery, the goal is data normalization. Strings are classified as date-time or numeric, and discovered as normalized NUMBERSTRING or DATETIMESTAMP data types.

Normalization is necessary in order to compare values. For instance, one column might contain a date in the form ‘9/7/09’, but another column might have a date in the form ‘07 September 2009’. Without normalization, we cannot determine if this is the same date. The same problem exists with numeric values.

For example, you might have selected Spanish in the Date and Number Format screen during installation, but your data could contain date, time, and number strings in both the Spanish and Italian locales. In this case, the scan formats need to be updated to correctly discover both Spanish and Italian dates and times simultaneously. Or, some of your data might be in a unique or unusual format not included in the default localized scan format file. In this case, you can “teach” this format to Discovery by modifying or creating a new scan format.

A pattern contains a positive subpattern and a negative subpattern, such as `#,##0.00;(#,##0.00)`. Each subpattern has a prefix, a numeric part, and a suffix. If there is no explicit negative subpattern, the negative subpattern is the localized minus sign prefixed to the positive subpattern. For example, 0.00 alone is equivalent to 0.00;-0.00. If there is an explicit negative subpattern, it serves only to specify the negative prefix and suffix; the number of digits, minimal digits, and other characteristics are ignored in the negative subpattern. For example, `#,##0.0#;(#)` has precisely the same result as `#,##0.0#;(#,##0.0#)`.

The number of pattern letters determines the format you use.
- Text strings with 4 or more pattern letters: use the full form.
- Text strings with 3 or fewer pattern letters: use the short or abbreviated form. If no short or abbreviated form exists, use the full form.
Numeric: use the fewest number of digits necessary. Shorter numbers are zero-padded to this amount. Year is handled specially; that is, if the count of ‘y’ is 2, the Year will be truncated to 2 digits. (e.g., if "yyyy" produces "1997", “yy” produces “97”.) Unlike other fields, fractional seconds are padded on the right with zero.

Text and numeric values with 3 or more pattern letters: use the full text form.

Text and numeric values with 1 or 2 pattern letters: use the numeric form.

You can change the scan formats used in a project, or the scan formats used in a particular data set. See "Changing a project's scan formats" on page 168 and "Change the Data Set Scan Formats" on page 168.

### The Home tab

The Home tab is the starting point for working in Discovery Studio. In this window, you create the project and identify the staging data source.

The following screen shows the Home tab when you first open Discovery Studio. By default, the Source Data Discovery (SDD) tab is selected. In this screen, two SDD projects have previously been created but no project is open. Discovery Studio is currently connected to the Discovery Server on the local host.

![Discovery Studio Home tab with no project open](image)

The following screen shows the Home tab after a Transformation Discovery (TD) project is opened. The screen is the same for SDD projects. Note that the Discovery screen tabs are now shown and can be selected. In addition, the currently open project is listed in the upper right corner of the screen. It is also listed in the title bar of the window. This information is shown in every Discovery screen.
The Project buttons allow you to create, open, close, edit, and delete projects.

TD projects and SDD projects have slightly different project summary information displayed in the Home tab.

The columns in the SDD project list are:
- Project Name
- Data Sets (Home tab)
- Activity
- Last Modified
- Notes

The columns in the TD project list are:
- Project Name
- Source
- Target
- Activity
- Last Modified
- Maps
- Notes

**Connecting to a different Discovery Server**

You can connect to a different Discovery Server while Discovery Studio is open. This allows you to work in projects stored in other Discovery Server repositories.

**Procedure**

1. Close any open project.
2. In the Home tab, click **Server Connect**.

---

*Figure 5. Discovery Studio Home tab with an open project*
3. Enter the hostname of a Discovery Server host in the network.
4. Enter the server port used by Discovery Server on the new host. The default port is 9090.
5. Click OK.

Results

The Discovery Studio Projects list now displays only the projects in the new Discovery Server repository.

The hostname and port number remain in the dropdown menu and you can select them in future Discovery Studio sessions.

Creating a project

The same procedure is used to create a TD project or an SDD project.

About this task

You need the following information to create a project:

- ODBC connection on this host to the staging data source
- User name and password for the staging data source

Note: If several projects will use the same staging data source, you can define and use a [Default Staging Data Source](#).

Procedure

1. Start Discovery Studio and connect to a Discovery Server. The Home window displays with the SDD project list shown by default.
2. Click **New Project**. The Create a new Project window appears. A new Transformation Discovery project is shown as an example.
3. Fill out the **General** fields.
   - **Type**: Select the project type, either [Transformation Discovery Projects](#) or [Source Data Discovery Projects](#). By default, the new project is the same type that was currently displayed in the Home window. You can select a different project type here and it will be saved in the correct list.
   - **Name**: It is a good practice to make the name descriptive of the data being processed or the desired output. Project names cannot be the same as any table or text file you will use in the project. Project names can contain alphanumeric characters (a-z, A-Z, 0-9, -, _ , spaces)
   - **Locale**: Select the locale that Discovery will use for the data in the project. The project, all of the databases in the project, and all of the Discovery hosts should use the same locale. You cannot change a project’s locale after the project is created.
   - **Notes**: Add notes on the project (optional).
   - **Use Case Sensitive Discovery**: This is on (selected) by default for [Transformation Discovery Projects](#) and off (unselected) by default for [Source Data Discovery Projects](#). Change it if required for the project. See [About Case Sensitivity](#).
   - **Use Byte Storage For String in Staging**: Select the appropriate option, depending on whether the data in the project is single-byte or multi-byte and how your staging data source was created (single-byte or multi-byte code set).
For all English-USA locales, the default is on (selected). Use this value for single-byte data. This can improve performance and reduce the storage space needed for the staging data source. Data is processed using byte semantics: for strings, Discovery uses VARCHAR2(N BYTE) for Oracle and VARCHAR(N) for DB2.

For all other locales, the default is off (unselected). Use this value for multi-byte data. The staging data source will require more storage space but multi-byte data will be correctly processed. Data will be processed using character semantics: for strings Discovery uses VARCHAR2(N CHAR) for Oracle and VARGRAPHIC(N) for DB2.

**Note:** If the DB2 staging data source was created with a single-byte code set, the Use Byte Storage For String in Staging selection must be manually unchecked. If this is not done, unrecoverable errors will be reported during profiling and staging.

4. If a default staging data source exists and you want to use it, leave the Use Default Staging option checked and proceed to Step 7.

   If there is no default staging data source or you want to use a different one, uncheck Use Default Staging and click Select. The Select Staging Data Source dialog appears.

5. Specify the staging data source as follows:

   **ODBC Connection Parameters**
   - **Connection Name:** Enter a meaningful name for this ODBC connection.
   - **ODBC Data Source:** Select the name of the staging data source from the dropdown list. The list displays all ODBC connections on this host. You are allowed to select an unsupported ODBC connection, but data discovery might fail or might be incorrect, or other problems might occur.
   - **User Name:** Enter a valid user name for this ODBC connection.
   - **Password:** Enter the password for the user name.
   - **Test Connection:** (optional) makes a test ODBC connection to the staging data source. If the connection is not successful, an error message describes the reason. Correct the information as necessary.

   **JDBC Connection Parameters**
   - **Database Server Name:** Enter the database server name for the JDBC connection.
   - **Database Instance Name:** (DB2 only) Enter the database instance name.
   - **Database SID:** (Oracle only) Enter the database system identifier (SID).
   - **Port Number:** Default staging data source port number for the selected database type. Accept the default (1521 for Oracle, 50000 for DB2), or enter a different value if the default is not correct for your installation.
   - **Connection URL:** This field is automatically populated as you fill out the previous fields. The URL must be correct in order for Discovery Server to connect to the staging data source. Changing information in the JDBC fields modifies the URL.
   - **Test Connection:** (optional) makes a test JDBC connection to the staging data source. If the connection is not successful, an error message describes the reason. Correct the information as necessary.
Notes: (optional) Enter a description or other comments about this ODBC connection.

6. Click OK to close the Select Staging Data Source window.

7. If desired, assign a password. This password will be required to open or delete the project. If you do not want a password, uncheck the Use Password box.

8. Click OK in the Create a New Project window.

Results

Discovery Studio creates and opens the project, then displays the Data Sets tab.

You can now Add Data Sets to Projects

Opening a project

You open a project in order to work in it.

About this task

Only one project at a time can be open in Discovery Studio. If you want two projects open at the same time on the same workstation, open a second instance of Discovery Studio and open the second project in the second instance.

You can queue tasks in a project and then close the project. The tasks will run even if you connect to a different Discovery Server or exit Discovery Studio.

After queueing tasks and closing a project, you can open a different project in the same Discovery Studio instance. You can work in the second project, including queueing tasks, while the first project’s tasks are running. Discovery Studio runs most tasks in serial order, so tasks in the second project will not run until the tasks in the first project are complete.

Procedure

1. Display the Home window.

2. Click the Transformation Discovery tab or the Source Data Discovery tab to display all of the projects of that type in the current repository.

3. Click on a project’s name in the list and then click Open Project, or just double-click the project name. If another project is already open in this Discovery Studio instance, Discovery Studio close the first project (asking you to save if necessary) and then opens the new project.

4. If the project requires a password, enter it at the prompt.

Results

Discovery Studio opens the project, then displays the last tab that was open for the project. The name of the open project is displayed in the upper right corner of the screen.

About locked projects

When you open a project, Discovery Server locks the project. Discovery Server removes the lock when you close the project, allowing other users to work in the project.

When you try to open a locked project, you have the option of opening it in read-only mode. However, you also have the option of breaking the lock and opening the project for editing. See Open a Project That Is Already Open.
Discovery Studio sends heartbeat messages at predetermined intervals to the Discovery Server. If Discovery Studio freezes or crashes, it stops sending heartbeats. It can also temporarily stop sending heartbeats while it is performing a memory-intensive processing activity. When Discovery Server does not receive an expected heartbeat, it waits for an additional specified period of time and then unlocks the current project, allowing you or another user to open the project again. The default heartbeat intervals and additional waiting period are usually long enough to allow Discovery Studio to complete a long processing task without Discovery Server accidentally unlocking the project.

If Discovery Server unlocks a project while Discovery Studio is running a task, the task still completes successfully and the results are displayed in Discovery Studio. You can continue working in the project. However, if another user opens the same project and cancels the task, you will have to rerun the task. Before you run a long task, it is a good practice to ask other Discovery Studio users to not open the project even if they see that it is unlocked.

The **HeartBeatInterval** parameter controls the frequency of Discovery Studio heartbeats. The **UISessionTimeout** parameter controls the length of time that Discovery Server waits after a missed heartbeat before unlocking the project. These values are set in the Discovery Server Configuration Utility.

### Opening a locked project

A project can only be opened by one Discovery Server instance at a time; two users cannot have the same project open at the same time, even on different workstations. If you try to open a project that is currently open in another Discovery Studio instance, you see a message asking you how to proceed. You can also see this message if Discovery Studio crashes or you shut down the system without first closing Discovery Studio or Discovery Server.

When Discovery detects that you are trying to open a project that is already open, you have three options:

- **Open the project in read-write mode.** This option immediately closes the other user’s access to the project and opens the project for editing in your Discovery Studio instance. The other user is not notified, and his or her changes since the last save are lost.
  
  Use this action with extreme care. Do not close another user out of a project unless you are very sure who the other user is and whether they are actively working in the project. However, if you are certain that the message is appearing because of a system crash or other application error and no one else is currently working in the project, it is safe to select this action.

- **Open the project in read-only mode.** You can view all tabs and most of the discovered data, but cannot execute any tasks or make any other changes to the project. The other user can continue working in the project without interruption.

- **Cancel the request.** The other user continues working in the project without interruption.

### Saving a project

Discovery Studio automatically saves the project at certain points in the discovery process, such as when you execute a task. You are also prompted to save any unsaved changes when you close a project and when you close Discovery Studio.

To manually save a project at any time, select **Project>Save**.
Closing a project

Closing a project makes the project available for other users to work in. Queued tasks will continue to run after you close a project. After closing a project, you can open another project or exit Discovery Studio. If you exit Discovery Studio without closing the current project, you are prompted to save the changes before exiting.

Procedure

1. Display the Home tab.
2. Click the Transformation Discovery tab or the Source Data Discovery tab, depending on the type of project that is open.
3. Select the open project in the list and then click Close Project.
4. If there are unsaved changes, you are prompted to save the changes. Discovery Studio then closes the project.

Deleting a project

Deleting a project removes all of the project information from Discovery Studio, as well as dropping all related logical tables and staging tables from the Discovery Server repository and staging data source. All sample sets, logical tables, and discovered results are permanently deleted. The source tables and files are not affected.

Procedure

1. Close the project.
2. Display the Home window.
3. Click the Transformation Discovery tab or the Source Data Discovery tab, to display the appropriate project.
4. Click on the project to be deleted.
5. Click Remove Project.
6. If the project is password-protected, a dialog will prompt you for the password. Enter the password and click OK.

Results

The project is immediately deleted.

Editing a project's properties

You can edit most properties of a project at any point in the discovery process. Some properties can be changed without any impact on the project. Changes to other properties can require you to rerun discovery steps or perform other actions. Several properties cannot be changed, such as the project type (SDD to TD, for example).

Procedure

1. Display the Edit Properties dialog of the project in one of the following ways:
   • In any window, click the project's name in the upper right corner of the screen.
   • Open the project and click Edit Properties in the Home window.
2. Change the properties as desired. If you cannot edit a property, it cannot be changed. If you change a property listed below, follow the action as noted.
**Case sensitivity**: if you have performed any work in the project, changing the case sensitivity can make the results incorrect. Rerun all Discovery steps to update the results.

**Use Byte Storage for String in Staging**: If you change this option, drop the profiling and staging tables and then rerun Column Analysis.

**Edit Scan Formats**: If you change this option, rerun all Discovery steps. See "Changing a project's scan formats." 

**Staging Data Source**: If you specify a new staging data source after you have imported any tables or text files, rerun all Discovery steps. See Change the Project's Staging Data Source.

**Variables**: You can define variables for a project. See Define Project Variables.

### Changing a project's scan formats

When the default scan formats in a project are not sufficient for your purposes, you can change the scan formats. This allows you to create custom format templates or mix different local formats.

#### About this task

Before you change the scan formats in a project, review the default scan formats and the patterns used in scan formats. See "About scan formats" on page 160.

#### Procedure

1. In the Home tab, double-click a project to open the Edit Project Properties window.
2. Click **Edit Scan Formats**. The Project Scan Formats window appears.
3. Click either **Number Formats** or **Date Formats**, depending on the type of scan format you want to create or change.
4. To edit an existing format, click in the appropriate **Format** cell. To create a new format, click the Add Format ( + ) icon to add a new blank row, then click in the blank **Format** cell.
5. Create the new format by typing directly into the cell, or by clicking the dropdown arrow and selecting the appropriate character from the dropdown menu. Add more characters as necessary to build the new scan format. You can edit the format directly in the **Format** cell. To see examples of positive and negative numbers using the current scan format, click outside of the dropdown menu.
6. When you are finished, click **OK** to close the Project Scan Formats window.
7. Click **OK** to close the Edit Project Properties window. The changes take effect with the next scheduled data type discovery task.

#### Change the Data Set Scan Formats

You can create additional scan formats for individual data sets. Data set scan formats are combined with project scan formats and used during column analysis.

#### About this task

Data set scan formats are combined with project scan formats and used during column analysis. Data set scan formats have the same requirements and restrictions as project scan formats.
Procedure
1. In the Data Sets tab, right-click a database connection and select Edit Scan Formats. The Scan Formats dialog for the selected data set appears. The list is empty until you create a data set scan format.
2. Click either Number Formats or Date Formats, depending on the type of scan format you want to create or change.
3. To create a new format, click the Add Format (+) icon to add a new blank row, then click in the blank Format cell. To edit an existing format, click in the appropriate Format cell.
4. Create the new format by typing directly into the cell, or by clicking the dropdown arrow and selecting the appropriate character from the dropdown menu. Add more characters as necessary to build the new scan format. You can edit the format directly in the Format cell. To see examples of positive and negative numbers using the current scan format, click outside of the dropdown menu.
5. When you are finished, click OK to close the Scan Formats window. The changes take effect with the next scheduled data type discovery task.

Changing a project's staging data source
If you have not yet imported any tables or text files into data sets, you can change the staging data source without affecting the project, regardless of whether the project used the default staging data source or a manually defined staging data source.

About this task
If you have already imported tables or text files, we do not recommend changing the staging data source. Doing so will cause Discovery Studio to lose access to all of the project's staged tables and text files, along with any existing profiling, target match, and data rule discovery results. Other application errors, including task failure and incorrect results, are also likely to occur. To avoid this, you must repeat all discovery steps, starting with Column Analysis, after changing a project's staging data source.

You can change a project's staging data source at any time using the Edit Project dialog. The new staging data source must be of the same database type and version as the existing repository and staging data source. For example, if the previous staging data source and repository were Oracle 10g R2, you cannot change to an Oracle 9 or DB2 staging data source.

Procedure
1. In the Edit Project dialog, click Select.
2. Enter the new staging data source as described in Create a Project.
3. Close the dialog.
4. If you have already imported tables or files into the project, Rerun Column Analysis and also repeat all subsequent discovery steps to recreate the data in the new staging data source.

Cleaning up the project's staging data source
You can drop tables in a project's staging data source from within Discovery Studio.
Procedure
1. In the Edit Project dialog, click **Clean Up Staging**. The Clean Up Staging dialog appears.
   Each table and text file in the project is shown, grouped by data set. If the project contains one or more unified schemas, the unified schemas are listed in a **Target Table** group.
2. Click a table to display its sample sets, then click each sample set to display the sample set's staging tables and indices.
3. Click on the checkboxes of the tables you want to drop. Clicking a sample set's checkbox automatically selects the staging tables and indices under the sample set. You can uncheck any items you do not want to delete.
4. When all of the desired items are checked, click **Clean Up**. The selected items are deleted from the staging data source.

About project variables
The Project Variables window shows the names and default values of the variables defined for the current project. You can add and delete project variables in this window.

The project variables defined in this window can also be used to **Create Scripts** for validation jobs for Exception Manager.

About importing projects
You can import a full project that you have previously exported from this version of Discovery. You can also import project objects, which are smaller exports containing only part of a project.

Staging Data Source and Repository Considerations in imported projects
When importing a project, the new repository and staging data source must be of the same database type and version as the repository and staging data source in the original project. For example, an export of a project created with a DB2 9.1 repository and staging data source can only be imported into a new project that also uses a DB2 9.1 repository and staging data source.

For more information, see **Staging Data Source**

Staging Tables in imported projects
Staging tables are not included in the exported file, and are not necessary for refining maps. To recreate the staging tables, rerun all discovery steps starting with Column Analysis.

Data Rules in imported projects
Data rule lookup tables are not included in the exported file. **Import Lookup Tables** into the project or rerun maps discovery to recreate data rules.

Importing a full project
All project information in the Export file is imported. You might need to recreate some objects if the new project needs them.
About this task

You can directly import projects that were created in the immediately previous version of Discovery. However, some objects (such as column matches or overlaps) might not be imported. The IBM InfoSphere Discovery Release Notes contain instructions for migrating to a newer Discovery release and for importing projects from the previous release.

Depending on the version, projects made in older releases of Discovery might not be supported. Contact Customer Support.

For more information about importing a full project, see Export Full Project.

Procedure

1. Close any open project.
2. In the Home window, click Import Project.
3. In the Browse for Folder dialog, select the folder containing the exported project, then click OK.
4. In the Import Project dialog, give the new project a name and specify a staging data source as described in Create a Project. The Notes and password are optional. Click OK when you are finished.

Note: The project type (Source Data Discovery or Transformation Discovery) is automatically detected from the exported file and cannot be changed.

5. A confirmation dialog appears when the import is complete. Click OK to close the confirmation dialog. Discovery Studio opens the new project and displays the Data Sets window.
6. If the statistics are stale, recreate the staging tables by rerunning all discovery steps starting with Column Analysis. See About importing projects for more information.

Importing TD project objects

Importing and exporting project objects is supported in TD projects only. You cannot import project objects exported from a previous version of Discovery, or exported from any other product.

About this task

You can import TD project objects that have been exported using the Export Project Objects procedure.

Procedure

1. Open an existing TD project or create a new TD project.
2. Select Project>Import>Project Objects.
3. In the Browse for Folder dialog, select the folder containing the exported objects, then click OK.
4. In the Import Objects dialog, highlight the objects you want to import into the project. Shift-click to select consecutive rows, or Control-click to select non-consecutive rows.

The Object Type column indicates whether an object is an option set, map, or data rule. Already Exists indicates whether an object with the same name already exists in this project.
When objects already exist, checking **Overwrite the Objects** replaces the old objects with the new ones. If this is not checked, the duplicate objects are not imported.

5. Click **OK**. A confirmation dialog appears when the import is complete, indicating how many objects were imported. This can take several minutes, depending on the size of the objects imported.

6. Click **OK** to close the confirmation dialog.

**Results**

Discovery Studio opens the new project containing the imported project objects.

---

**Create Data Sets**

A *data set* is a collection of database tables and text files to be processed or mapped.

**Overview**

Discovery allows you to create two kinds of projects.

- **Transformation Discovery Projects** contain one *source data set* and one *target data set*, with the goal of mapping the source data set to the target data set. A data rule data set is included by default, but it is not necessary to use it.
- **Source Data Discovery Projects** contain at least one data set. The data sets are not identified as source or target because SDD projects identify overlaps, not mappings. SDD projects do not use data rule data sets.

After creating a project, you populate the data sets with the appropriate database tables and text files. Each data set can contain as many database connections as needed. You can import multiple database tables from each connection into each data set. There is also no limit to the number of text files that can be imported into each data set.

Any data set (including the data rule data set) can include text files and tables from any supported database. For example, a data set can contain Oracle, DB2, and Sybase tables, along with text files. The list of supported database types is available from Customer Support.

**How Discovery Handles Database Tables and Text Files**

When you import a database table into a data set, Discovery reads the metadata and creates a staging table in the staging data source containing the metadata. At a later point in the analysis process, Discovery reads and stages a copy of the entire table in the staging data source

Text files that you import are read and converted into staging tables that are also saved in the staging data source. After this, text file data sources are treated exactly the same as database table data sources.

**Note:** Discovery never modifies the actual user data tables or text files. All actions are performed on the tables in the staging data source.

**Differences in TD and SDD Projects**

The Data Sets window is slightly different in TD projects and in SDD projects.
SDD projects contain one data set by default, *Data Set 1*. The **Overlaps**, **Unified Schema**, and **Unified Schema PF Keys** tabs are shown.

TD projects contain three data sets by default: *Source Data Set*, *Target Data Set*, and *Data Rule Data Set*. The **Target Matches** and **Maps** tabs are shown. A Data Rules menu also appears in the menu bar.

In all other ways, the Data Sets window is the same for both projects. Some examples in this section show a TD project, and other examples show an SDD project. The instructions are the same for both types of projects.

The empty Data Sets tab for SDD projects is shown below.

The empty Data Sets tab for TD projects is shown below.

**Source and Target Data Sets in TD Projects**

TD projects have three data set tabs with the default names of *Source Data Set*, *Target Data Set*, and *Data Rule Data Set*. TD projects require that you populate at least one data set, either the source data set or the target data set. Unused data sets can remain empty and do not need to be deleted from the project.

The source and target data set tabs appear in the **Column Analysis**, **PF Keys**, and **Data Objects** tabs, allowing you to perform tasks on each data set separately in those steps. In the **Target Matches** and **Maps** tabs, you perform actions on objects that contain columns and tables from both data sets.

**Using All Data Sets or One Data Set:**
Not all TD projects require both a source and a target data set. For example, if the goal is discovering the relationships (data objects) within a set of tables and you don’t need maps, only one data set is required. That data set is not considered either a source or a target.

When a project does not use a source and target, you import the database tables and text files into one data set and leave the other data set empty. It does not matter whether you use the source or the target data set. Ignore any errors or messages regarding the empty data set.

The data rule data set is only used for lookup tables required by transformations that involve data rules. If your project does not use lookup tables, you can leave this data set empty.

**Determining Which Data Set is the Target:**

Before you import any user data files into the data sets, determine which files are the target and which are the source. Discovery Server treats the source and target data sets very differently. When deciding which data set should be the source and which should be the target, remember this:

*The target data set is the focus of the mapping project.* In other words, the goal of all analysis and discovery tasks is to transform the source data set into the target data set.

For each column in the target data set, Discovery Server tries to discover a transformation that can be applied to one or more source columns to convert the data in those source columns to the data in the target column.

All discovery and analysis actions are performed with this objective. As Discovery Server discovers increasingly stronger relationships (primary-foreign keys, data objects, and target matches) within and between the source and target data sets, it filters out more and more source columns that have no relevance to any target column. By the time Discovery Server creates maps, it is only performing discovery and analysis tasks on source columns that have a proven relationship to one or more target columns.

**Data Sets in SDD Projects**

SDD projects contain one data set by default, *Data Set 1*. You can [Add Data Sets to Projects](Projects) as needed. You must populate at least one data set in an SDD project.

Each data set's tab appears in the Column Analysis, PF Keys, and Data Objects tabs, allowing you to perform tasks on each data set separately in those steps. In the Overlaps, Unified Schema, and Unified Schema PF Keys tabs, you perform actions on objects that contain columns and tables from all data sets in the project.

**Note:** All data source names within a project must be unique.

**About Text Files**

Text files contain no metadata. When Discovery imports text files, it assigns the VARCHAR type to all fields. During the Column Analysis step, Discovery examines all user data files in the data source, both database tables and text files, and converts VARCHAR datatypes into other datatypes as appropriate.
Tables that Discovery creates from text files are handled and analyzed by Discovery just like database tables. In fact, from this point on, text file data sources appear and behave in Discovery just like database data sources.

**Access to Text Files**

Text files are specified in Discovery Studio using the full directory path. You cannot use mapped network drive names to specify a text file.

Text files used in any data set can be located on any machine, as long as three conditions are met:

- In order to add a text file to a data set (or re-import the text file), the Discovery Studio host must be running in a Windows session where the logged-on Windows user has read permission for the text file directory. After a text file is added to a project, the project can be opened and worked on by any Discovery Studio in the system, whether or not the logged-on Windows user has permission for that text file directory.
- The logged-on Windows user on the Discovery Studio host must be able to access and must have at least read permission for each text file. If the logged-on Windows user does not have read permission, an error will occur during any discovery or processing tasks involving that file.
- The Discovery Engine Service Logon Account must have access privileges to the file. The default logon account has local privileges only. To change the logged-on Windows user, see [Change the Discovery Engine Service Logon Account](#).

**About Logical Tables**

Logical tables allow you to standardize column names, transform data to standardized semantics, and perform analysis on data that is tailored to a particular business need.

When you import a database table or text file into a data set, Discovery creates a logical table from that physical table or file. It also creates the Full Set sample set from that logical table, not from the physical table or file. All processing is performed on the logical table, not on the physical table.

You can create additional logical tables for each physical table or file. (Before you do, see [Modify a Logical Table](#)) In each logical table, you can define joins and where clauses using other imported physical tables and files. Each logical table can have multiple sample sets that further filter the data, but only one sample set at a time can be the primary for that logical table. If you have defined more than one logical table for a physical table or text file, Discovery treats each logical table as a separate table during processing.

The joins and where clauses in a logical table can include physical tables and files from any data set in the project. The physical tables and text files referenced in a logical table do not need to be from the same database connection or be of the same database type.

There is a [Maximum Number of Logical Columns](#) that a project can contain.

**About Sample Sets**

You can add, delete, and modify sample sets, and designate a different primary sample set, at any time in the analysis process. However, you may need to re-run some steps.
When you import a physical table or text file, Discovery automatically creates a logical table for it and then creates a sample set named Full Set from the logical table. The Full Set sample set contains the full contents of the logical table, and is designated by default as the primary sample set.

Discovery performs all processing and analysis tasks on each logical table's primary sample set. You can create additional sample sets for each logical table containing different subsets of the data in the logical table, and designate any one of the additional sample sets as the primary sample set. If you create additional logical tables for a particular physical table, each logical table has its own primary sample set that is used in the processing and analysis tasks.

**About Importing Foreign Keys**
When you import a table into a data set, Discovery can import foreign keys between the new table and an existing logical table in the same data set.

The foreign keys are imported if the following conditions are met:
- The table names, number of columns, and column names in each table have not changed since the foreign key was created.
- The logical table has the same table name, column names, and number of columns as its corresponding physical table in the data set.
- You have not created any joins on the logical table.
- You do not change the name of the new table when you import it.

For example, assume TABLEA and TABLEB have a primary-foreign key relationship, where TABLEB contains the foreign key. After you import both tables, in any order, Discovery recognizes the foreign key between TABLEB->TABLEA.

If you reimport TABLEB into the data set, it is automatically renamed to TABLEB1 because TABLEB already exists in the data set. Discovery recognizes the foreign key relationship between TABLEB1->TABLEA.

If you now reimport TABLEA into the data set, it is automatically renamed to TABLEA1. Discovery does not recognize the foreign key relationship between TABLEB->TABLEA1 because the key is to TABLEA, not TABLEA1.

**About Exporting Projects**
Discovery allows you to export projects in a variety of formats for import into different third-party products. Some products have certain requirements related to imported data. For example, the number of rows or database connections in a data set may be limited, certain data types may be converted, text files may not be included, and so on.

When a project will be exported for use in another product, make sure you accommodate all third-party requirements. Read this introduction thoroughly, and also read the relevant sections related to the particular export in Export Data.

**About CWM (Common Warehouse Metamodel) Files**
You can import CWM (Common Warehouse Metamodel) XML files into a project. CWM files can be imported as physical tables, as logical tables, or as a unified schema target table.

For instructions, see the following sections:
- Import CWM Files as Physical Tables
- Import CWM Files as Logical Tables
About Classification

In the Data Sets window, you classify each data set as either Operational (OLTP, the default) or Data Warehouse. Classification types are only used during PF Key discovery, but they directly affect the structure and quality of the data objects that Discovery discovers for that data set. Meaningful data objects are the basis for reliable and accurate results later in the analysis process.

During PF Key discovery, Discovery assigns a classification to each logical table in the data set. The available logical table classification types are based on the data set classification type. For example, logical tables in an Operational type data set can be parent entities or child entities, while logical tables in a Data Warehouse type data set can be facts or dimensions. Data Set Classification describes available types.

Each PF Key relationship between the logical tables in each data set is also given a classification that best describes the nature of the relationship. This is also based on the data set classification type. PF Key Relationship Classification describes the available PF Key relationship types.

Data Set Classification:

Select the data set classification based on the structural type of the physical database tables in the data set, and your project goal. Data sets in the same project can have different classifications.

For example, if you are migrating a Data Warehouse database to an Operational database, classify the source data set as Data Warehouse, and the target data set as Operational.

If you are not sure which classification to use, keep the default classification of Operational (OLTP).

Operational (OLTP):

Operational (Online Transaction Processing, or OLTP) is the default classification. This is an entity relationship data model with application-oriented design. Traditional applications development environments use OLTP.

Data Warehouse (Star Schema or OLAP):

Data warehouse, also called star schema or online analytical processing (OLAP), is a method of storing multidimensional data. It uses a single fact table with a compound primary key containing one segment for each dimension.

PF Key Table Classification:

During PF Key discovery, Discovery assigns a classification to each logical table (database table or text file) in each data set. The type of table classification depends on the data set classification.

Tables in an Operational Data Set:

Logical tables in an Operational data set are one of the following types:
• **Root entity**: A root table that defines entities in the database. For example, a root entity table named ORDERS may have a primary key to a child entity table named ORDER_ITEMS.

• **Child entity**: A child table that provides additional information about items defined in root entity tables. For example, a child entity table named ORDER_ITEMS may have foreign keys to a root entity table named ORDERS.

• **Cross-Reference (or Lookup)**: A table whose primary key is formed by foreign keys from two tables; a table that creates many-to-many relationships between two tables. For example, a cross-reference table named PROD_BY_SUPPLIER may have a SUPPLIER_ID column that is a foreign key to SUPPLIERS, and a PROD_ID column that is a foreign key to PRODUCTS.

• **System**: This classification is not discovered by Discovery, but is manually assigned by the user in the PF Keys window.

• **Unclassified**: The initial classification of all tables in the data set before Discovery discovers a classification.

**Tables in a Data Warehouse Data Set:**

Logical tables in a Data Warehouse data set are one of the following types:

• **Dimension**: A root table that provides additional information about an item in the fact table; a table with many primary keys that point to foreign keys in a fact table. For example, a dimension table named CUSTOMERS may contain the name and other information about each customer ID listed in the fact table SALES.

• **Fact**: A root table that contains individual instances of some action, such as a purchase or a shipment; a table with many foreign keys that constitute its key. For example, a fact table SALES may have primary keys to the dimension tables PRODUCTS, CUSTOMERS, and CHANNELS, and to the attribute table STORE_LOCATION.

• **Reference**: A table with primary keys that point to foreign keys in a dimension table. For example, a reference table named SUPPLIER may have a column named PROD_ID that is a foreign key to the PRODUCTS dimension table.

• **Cross-Reference (or Lookup)**: A table whose primary key is formed by foreign keys from two tables; a table that creates many-to-many relationships between two tables. For example, a cross-reference table named PROD_BY_SUPPLIER may have a SUPPLIER_ID column that is a foreign key to SUPPLIERS, and a PROD_ID column that is a foreign key to PRODUCTS.

• **System**: This classification is not discovered by Discovery, but is manually assigned by the user in the PF Keys window.

• **Unclassified**: The initial classification of all logical tables in the data set before Discovery discovers a classification.

**PF Key Relationship Classification:**

Discovery recognizes three classifications of PF Key relationships. Regardless of the Data Set Classification and PF Key Table Classification, each PF Key is classified as one of the following:

• **Parent-child**: The parent table contains an entity, and the child table contains additional attributes of the entity.

• **Reference**: The child table contains an entity, and the parent contains reference attributes for the entity. A reference relationship is typically one-to-many or one-to-one.
• *None:* The relationship has not yet been defined. This is the default type for defined PF Keys.

**The Data Sets Window**

The name of the open project is shown in the upper right of the screen and in the title bar.

**About this task**

In the left-hand pane, you rename the selected data set and add [Notes] (if desired), and set the classification type.

**Note:** All data source names within a project must be unique.

You can rename a data set at any time. If you have already begun processing data (run Column Analysis or any other step), renaming the data set will cause Discovery to save the project, which updates the staging data source and other resources.

The pane on the right allows you to add connections and data sources (physical tables and text files) to the selected data set. For each physical table or text file, Discovery creates a logical table and lists it in the Logical Tables section. A Full Set sample sets is created for each logical table and listed with the logical tables.
To begin creating an SDD project, see Add Data Sets to Projects.

To begin creating a TD project, see Name and Classify the Data Set.

**Import Objects Pane**

The Import Objects pane lists all of the database tables and text files imported into the selected data set, along with the sample sets and logical tables for each.

A sample populated Import Objects pane is described below.
When you create a database connection, the connection is added to the **Database Connections & Tables** area of the Import Objects pane. Each physical database table you add from that connection is listed under the connection name.
The physical table or text file is listed in the **Database Connections & Tables** or **Text File Formats & Files** list with a _PT suffix. This identifies it as a physical table or text file. The **Logical Tables** list contains the automatically-created or manually defined logical tables and sample sets.

When you import a physical text file, all of the settings you specify for that file during import (delimiters, column separators, escape strings, error handling, etc.) are saved as a format in the **Text File Formats & Files** section. The text file itself is listed as the first entry under that format. Additional text files that use the same settings and that are in the same format can be imported directly into that format, and are combined into the same Full Set sample set. For more information, see [Combining Multiple Text Files into One Table](#).

Discovery automatically creates a logical table for every imported physical table and text file. You can create additional logical tables from a physical table or file. Before you do, see [Modify a Logical Table](#).

Discovery also automatically creates a default sample set (named Full Set) for each logical table, and for each additional logical table you create. The default sample set contains the entire contents of the logical table. You can create and customize additional sample sets for each logical table, but only one sample set at a time can be marked as primary. The primary sample set for each logical table is marked with a green dot.

For more information on logical tables, see [About Logical Tables](#) and [About Logical Tables](#). For more information on sample sets, see [About Sample Sets](#) and [About Sample Sets](#).

### Add Data Sets to Projects

Data sets can only be added in SDD projects.

**About this task**

SDD projects require that you have at least one populated data set. One data set tab is present by default. Add as many additional data sets as needed.

**Note:** To delete a data set, right-click on the tab and select **Delete Data Set**. The data set and all associated connections are deleted from Discovery. (The actual source files are not affected.) After deleting a data set, re-run all discovery steps.

To add a data set to a project:

**Procedure**

1. Right-click on an existing data set tab.
2. Select **Add Data Set** in the popup. A new numbered data set tab appears at the end of the current list of tabs.

**Results**

You can add additional data sets now, or **Name and Classify the Data Set** and add more data sets later.

**Name and Classify the Data Set**

It is useful, but not necessary, to **Name or Rename the Data Set**. You can use the default names, but giving each data set a meaningful name may make subsequent steps easier. Verify that the classification is correct and change it if necessary.

At any time, you can change the name of a data set without affecting any data discovery that has been completed. If you have already begun processing data (run Column Analysis or any other step), renaming a data set will cause Discovery to save the project, which updates the staging data source and other resources.

**Note:** If you **Add Data Sets to Projects** or change the contents of a data set, you need to re-run all discovery tasks.

You can also **Change the Data Set Classification** at any time. If you do so, you must re-run all discovery steps.

**Note:** The screen shots in this procedure show an SDD project. The process is the same for a TD project.
Name or Rename the Data Set
To change the name of a data set:

Procedure
1. Click the desired data set tab to select it.
2. Click Rename.

If you have already run Column Analysis or any other step, a dialog may appear stating that the project will be saved. Click Continue in this dialog to rename and save the project, or Cancel to cancel the rename task and not save the project.
3. In the **Name** field, enter a meaningful name for the data set, up to 30 alphanumeric characters including spaces, dashes and underscore. It is a good practice to include a descriptive word in the name, such as a client or department name, year, region, or any other identifying information. Data set names are used to identify this data in every Discovery Studio screen and in all Exception Manager validation results.

**Results**

**Note:** Data set names must be unique within the project.

If the data set name is invalid, a flashing red ! appears. Change the data set name to use only valid characters.

Proceed with [Change the Data Set Classification](#)

**Change the Data Set Classification**

**About this task**

You can change the **data set Classification** type, if it is not correct. For more information, see [About Classification](#)

**Note:** If you change a classification type after you have performed any processing tasks, you must re-run all steps.

**Procedure**

In the **Classification** field, verify that the selected classification type is correct. Change it if necessary. The options are:

- **Operational (OLTP)** (default)
- **Data Warehouse** (star schema)
Results

You are now ready to populate the data set using one or more of the following procedures. It does not matter which data set you populate first.

- **Import Database Tables**
- **Import Text Files**
- **Import CWM Files as Physical Tables**

**Import Database Tables**

Database tables can reside in any schema, and a data set can contain physical tables from different schema as long as a JDBC connection is available for each schema.

**Create Database Connection**

**About this task**

Discovery (TD) projects and Source Data Discovery (SDD) projects use JDBC connections to database data sources. The screen shots in this procedure are from an SDD project, but the process is the same for a TD project.

**Procedure**

1. To add the first physical table in a data set, click the Database Connection radio button and then click Click here to add a new connection!
Note: After you enter the first connection, this option disappears. To add more connections, use one of the following methods:

- click **Database Connections and Tables**, then click +.
- right-click **Database Connections and Tables**, then select **Add a new ODBC Data Source**.
  The *Create Connection* dialog appears.

2. Enter a meaningful connection name. This name will appear in the **Database Connections & Tables** list.

Note: Each connection name in the project must be unique. For example, you cannot use the same connection name in two separate data sets in the same project.

3. For TD projects, enter information in the **ODBC Connection Parameters** section. You can click **Test Connection** to verify the connection. Make sure the host is available over the network.
   a. **ODBC Data Source**: Select the appropriate ODBC data source from the dropdown list. This list shows all ODBC connections on the current host.
   b. **User Name** and **Password**: Enter a valid user name and password for this ODBC data source.

4. For TD and SDD projects, the JDBC connection fields do not appear in exactly the same order. Enter the JDBC information in **JDBC Connection Parameters** for TD projects, and in **Connection Parameters** for SDD projects. You can click **Test Connection** to verify the connection. Make sure the host is available over the network.
   a. **Database Type**: (SDD only) The type of database containing the source or target data.
   b. **Database Server Name**: Enter the database server name for the JDBC connection.
   c. **Database Instance Name** or **Database Name**: (DB2 only) Enter the database instance name.
   d. **Database SID**: (TD only) Enter the database system identifier (SID) (Oracle only)
   e. **Specify Port** (MSSQL only): Leave checked to use static ports, and then specify a port in the **Port Number** field. Uncheck when using dynamic ports.
f. **Port Number**: Default staging data source port number for the selected database type. Accept the default (50000 for DB2, 1521 for Oracle), or enter a different value if the default is not correct for your installation.

g. **User Name and Password**: (SDD only) Enter a valid user name and password for this JDBC data source.

h. **Connection URL**: This field is automatically populated as you fill out the previous fields. The URL must be correct in order for Discovery Server to connect to the staging data source. Changing information in the JDBC fields modifies the URL.

5. (optional) In the **Notes** field, enter a meaningful description for this connection. This is helpful if you are creating many connections.

6. To have Discovery perform bulk read-write operations to the staging data source, leave the **Use Turbo Mode** option checked. This is faster and better for most projects.

   If you uncheck this option, data is read row by row and column by column. This is more thorough, but performance is much slower and invalid data are replaced by null values.

7. Click **OK** to close the Create Connection dialog.

**Results**

Continue with the next step, **Import Physical Tables/File Formats**

**Import Physical Tables/File Formats**

**About this task**

**Note**: There is a [Maximum Number of Logical Columns] that can be imported into a project.

**Procedure**

In the Import Objects list, right-click on the connection you just created and select **Import Tables/File Formats**. (Or, click on the connection and then click the icon.

**Results**

**Note**: In SDD projects, the **Reload Data** option is not available.
Continue with the next step, **Import Table: Search Criteria**

**Import Table: Search Criteria**

**Procedure**

In the *Import Table Wizard*, set the search criteria for displaying a list of physical tables, then click **Next>>**. You will choose the physical tables or views to import from this list.
To limit the search to physical tables or views, select one or both of the following options (checking both options searches for both types):

- **Include Tables** limits the search to only physical tables only.
- **Include Views** limits the search to views only.
- **Ignore Empty Tables** limits the search results to only tables or views with at least one row of data. However, this can greatly increase the search time, depending on the number and size of tables or views in the data source.

After selecting tables or views, you can further refine the search criteria using the following options:

**Note:** When searching for a physical table or view that contains a period (.) in the name, enclose the table or view name in quotes.

- **All Tables** expands the search results to include all tables or views in all schemas in the data source. This may take a considerable amount of time and display more tables or views than you want to import.
- **Search Tables** allows you to search all schemas for tables or views containing a particular string.
  - To find all tables or views containing a particular string, enter the string in the **Table Name** field. Do not include a schema name or a separator ( . ).
  - To search a particular schema for all tables or views starting with a specific string, check **Search Tables** and use the format `[schema].[table]` for the string. You can widen the search by entering only the last few characters of
the schema name and the first few characters of the table or view name. For example, CE.REP will discover SOURCE.REPORT along with HQCE.REPORTS_2007 and SOURCE.REPDATA.

- To display all tables or views in a particular schema, use the format [schema]. and include the separator (. ) after the schema name. You can widen the search by using only the last few characters of the schema name.

  - **Search Columns** allows you to search all schema for columns containing a particular string. In order to be listed in the results, a table or view must have a column containing this string. Enter the string in the **Column Name** field. Do not include a separator (. ).

After specifying the search criteria, you can use filter files to further restrict the tables or views listed in the results. You can create as many filter files as needed, but can only specify one filter file per search.

**Note:** To use **Filter Search Results**, create a text file (filter file) containing one table or file name per line with a hard return at the end of each line. Do not include any tabs or wildcards. Leading or trailing spaces in the filter file are ignored. The filter file is not case-sensitive.

- **No Filter** clears any previously-set filter. Search results are not filtered.
- **Exclude** allows you to select a text file containing the names of tables or views to be excluded from the results. Tables or views that exactly match any entry in the file are filtered out of the search results. The filter is not applied to schema names.

  - **Include Only** allows you to select a text file containing the names of permitted tables or views. In order to be shown in the search results, a table or view must exactly match a string in the file. The filter is not applied to schema names.

Discovery Studio connects to the data source and displays a list of physical tables or views meeting the search criteria, along with the total number returned and the number currently selected for import.
Import Table: Select Tables

Select the tables you want to import. Press Next to view or edit the table names.

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Table Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HQ_ADDRESSES</td>
<td>SOURCE_SCHEMA.HQ_ADDRESSES</td>
<td>Table</td>
<td>[TABLE]</td>
</tr>
<tr>
<td>HQ_DEPARTMENTS</td>
<td>SOURCE_SCHEMA.HQ_DEPARTMENTS</td>
<td>Table</td>
<td>[TABLE]</td>
</tr>
<tr>
<td>HQ_EMP</td>
<td>SOURCE_SCHEMA.HQ_EMP</td>
<td>Table</td>
<td>[TABLE]</td>
</tr>
<tr>
<td>HQ_SALARIES</td>
<td>SOURCE_SCHEMA.HQ_SALARIES</td>
<td>Table</td>
<td>[TABLE]</td>
</tr>
<tr>
<td>HQ_STORES</td>
<td>SOURCE_SCHEMA.HQ_STORES</td>
<td>Table</td>
<td>[TABLE]</td>
</tr>
</tbody>
</table>

- **Display Name** is the table or view name only.
- **Table Name** is the full table or view name, including schema.
- **Type** indicates whether it is a **Table** or a **View**.
- **Count** displays the number of rows in each table. This column only appears if you checked **Ignore Empty Tables** in the previous screen.
- **Description** contains any description associated with the table from the database application.

Results

Continue with the next step, [Import Table: Select Tables](#).

**Import Table: Select Tables**

**Procedure**

1. Select the physical tables you want to import in one of several ways:
   - Click on a single table to select it. This allows you to look at the data before import by clicking **Preview Table**.
   - Select several nonadjacent tables using CTRL + click.
   - Select a range of adjacent tables using SHIFT + click.
   - Click **Select All** to select all tables in the list.
     Or, click **Back** to return to the previous dialog and specify different search criteria.
2. Import the physical tables, or edit the names before importing them.
   - To import the selected tables as shown, click **Finish**.
   - To edit any of the table names, either to make them more descriptive or avoid having duplicate table names in the project, click **Next**.
Results

Continue with the next step. Import Table: Table Names

**Import Table: Table Names**

**Procedure**

1. Click in the appropriate row in the *Table Name* column and edit the physical table names as necessary. When you are done, click *Finish*, or click *Back* to return to the *Select Tables* dialog.

   **Note:** Changing the table name here does not change the actual name of the physical table in the database. It changes the name of the staging table that Discovery creates in the staging data source, and the name of the logical table created from the physical table.

The Import Tables Wizard closes. The *Connections* list now shows the connection name along with the name of each physical table you have imported.
2. Repeat the [Import Database Tables] procedure for all additional required physical database tables in the remaining data sets.

**Importing from Guardium**
You can import database table and column definitions from Guardium®.

**Procedure**
1. From the menu bar, click **Project > Import > Import from Guardium**.
2. From the browser window, select the valid exported *.csv file.

**Results**
The database definitions are now available to work with in InfoSphere Discovery.

**Import Text Files**

During the import process, you specify the internal structure of the text file. This allows Discovery to convert the contents to an appropriate table format.

**Note:** The Maximum lines in text file import wizard option determines how many rows are read in the text file for profiling purposes. If a value in a row beyond the last line has a different characteristic, an error may occur according to the error handling set in the **Table Properties** screen of the text file import wizard, and data preview may not be accurate. This error will be resolved when Column Analysis is run. For example, if the maximum is 1000 lines, the column width is set during import to the longest value in those 1000 lines. If a value in line 1050 is longer than any previous value in that column, an error occurs. Line 1050 is treated according to the error handling selection (treated as null, etc.). However, when you run Column Analysis, Discovery scans the entire text file and sets the maximum column width to the length of the value in 1050.

You can import delimited and positional physical text files. These text files can be located anywhere in the network as long as:

- The user logged on to the Discovery Studio machine has Read permission for the text file directory.
The logged-on Windows user on the Discovery Studio host must be able to access and must have at least read permission for each text file.

You specify the full machine name (UNC name) in the filepath, such as \orion\Data\Upload\SourceFiles\05232007.txt. Do not use mapped drives, such as Z:\SourceFiles\05232007.txt.

Discovery reads the first 1000 rows of the text file and sets the column width to the width of the longest value found, up to 512 characters. If longer values are encountered in subsequent rows, Discovery increases the column width until it is a maximum of 512 characters. Values longer than 512 characters are truncated.

Select the Text File
To import a physical text file:

About this task

Procedure
1. Click the Text File Format radio button and then click Click here to add a new connection!

Note: After you add the first text file format, this option disappears. To add more connections, use one of the following methods:

   - click Text File Formats & Files, then click .
   - right-click Text File Formats & Files, then select Import File Formats.

2. Navigate to the physical text file you are importing and click Open.

   The Add Text Table wizard opens with the file displayed in the preview pane. If necessary, you can scroll horizontally and vertically to view the full data. You can also click Text Preview to display the source file data.
For delimited text files, continue with **Add Delimited Text File: File Type**.

For fixed width text files, continue with **Add Fixed Width Text File: File Type**.

### Results

#### Text Preview

You can click **Text Preview** to display the physical text file data. The preview does not show any of the formatting or other changes made in the file import wizard screens.

**Add Delimited Text File: File Type**

**About this task**

Complete the [Import Text Files] procedure before performing this procedure. The **Add Text Table** wizard is displayed.
Procedure

1. Select the **Delimited** text file type.

2. Select the appropriate **Row Delimiter** and **Escape String** from the dropdown menus. If the correct row delimiter or escape string is not listed, select **Other** and enter the appropriate ASCII string in the empty text box.

   **Note:** Some ASCII characters cannot be typed directly into the text box. In this case, click **Text Preview** to display the file, then cut and paste the character into the text box.

   The preview pane immediately displays the results of applying the selected values. Adjust the values until the table layout is correct. You can also click **Text Preview** to display the source file data.

   - **Row Delimiter** is the string that indicates breaks between one row and the next. This is a required field.
   - **Escape String** specifies one or more escape characters which mark a special character (such as the column or row delimiter) as ordinary text. This only applies to characters that have already been designated as special delimiters.

3. In **Heading Line**, enter the line number of the line containing the file headings. This is usually the first line in the file (line 0). Discovery will disregard the specified line when converting the file to a table.

   **Note:** When a heading line is specified, any value in that line that begins with En (where n is any digit) is automatically renamed to Jr. This avoids possible interpretation of En as a floating point value instead of as a column name.
If a file has no heading lines, uncheck the **Heading Line** box.

4. In **Skip Lines**, enter the line numbers of any lines to be skipped when Discovery converts the physical file to a table and creates the logical table. Specify lines in the following ways:
   - A single number (0)
   - A hyphenated range of rows, inclusive (10-20)
   - Multiple entries separated by commas (0, 5, 10-15, 18, 27-40)

   **Note:** The first line in the text file is Line 0.

**Results**

Continue with the next step, **Add Delimited Text File: Delimiters**

**Add Delimited Text File: Delimiters**

**Procedure**

In the next screen, select the appropriate **Column Delimiter** and **Text Delimiter**. If the correct character is not listed, select **Other** and enter the appropriate ASCII string in the blank field. The preview pane immediately displays the results of applying the selected string.

You can click **Text Preview** here also to preview the physical text file.

**Note:** Misplaced column delimiters within the text file can create extra columns. When this happens, the column(s) appear at the far right with the name **COLUMN_ n**, where **n** is the column number. If any data is misaligned, exit the wizard, correct the file, and restart the wizard from the **Import Text Files** step.
Column Delimiter is the string that Discovery uses to separate columns. The column delimiter you select here will be used for all future physical text file imports until you manually select a different delimiter.

Text Delimiter is a character that indicates the following string is a data value and not a special character, command, etc. The text delimiter you select here will be used for all future physical text file imports until you manually select a different delimiter.

Results

Continue with the next step, Add Delimited Text File: Column Properties

Add Delimited Text File: Column Properties

Procedure

One by one, click in each column and review the column's properties in the Column Properties fields. The first column is selected by default.
1. **Column Name** matches the column header shown in the preview pane. If there is no header line, the name is COLUMN_0. You can edit the **Column Name** field. The preview pane displays the new column name when you click out of the field.

2. **Skip Column** determines whether Discovery will skip (not include) this column when converting the physical file to a table and creating the logical table.

3. **Truncate Whitespace** determines whether Discovery will remove empty spaces at the beginning or end of this column when converting the physical file to a table and creating the logical table. If this is not checked, empty spaces are converted to spaces.

   - **Null Value Handling** determines how a null value detected in this column is handled when Discovery converts the physical file to a table and creates the logical table.
     - **TreatAsNull** converts it to a null value.
     - **TreatAsEmpty** converts it to an empty cell (do not use this option with numberstring columns).
     - **TreatAsError** causes Discovery to execute the **Error Handling** action selected in the next step.

**Results**

Continue with the next step, **Add Delimited Text File: Table Properties**.
Add Delimited Text File: Table Properties

Procedure

1. Verify that the table properties are correct.

   - **Table Name** is the name of the imported physical text file. You can edit the filename here. The name of the original text file is not changed, but Discovery will use the new name when it saves the text file as a logical table.
   - **Description** is an optional field that can be useful when importing a number of similar physical text files.

2. Click **Finish**.

   The physical text file is added to the Data Sets window in the **Text Files** list. A logical table and a sample set representing this file are also added in the Logical Tables section. Although it is listed under **Text Files**, it has been converted to a table, and Discovery will treat it as a table in all further processing and analysis actions.

Results

To add another physical text file to this table, see **Combining Multiple Text Files into One Table**. The new text file must be in exactly the same format as the first text file and must use the same processing options.

To import a different physical text file, repeat the **Import Text Files** procedure.
Add Fixed Width Text File: File Type

About this task

Complete the Import Text Files procedure before performing this procedure. The Add Text Table wizard opens and displays the selected physical text file. You can click Text Preview to view the text file.

Procedure

1. Click the Fixed Width text file type.
2. Select the appropriate RowDelimiter and Escape String from the dropdown menus. If the correct row delimiter or escape string is not listed, select Other from the appropriate dropdown menu and enter the appropriate ASCII string in the text box.

Note: Some ASCII characters cannot be typed directly into the text box. In this case, click Text Preview to display the file, then cut and paste the character into the text box.

The preview pane immediately displays the results of applying the selected values. Adjust the values until the table layout is correct.

- RowDelimiter is the string that indicates breaks between one row and the next.
- Escape String defines one or more escape characters which mark a special character (such as the column or row delimiter) as ordinary text. This only applies to characters that have already been designated as delimiters.
3. In **Heading Line**, enter the line number of the line containing the file headings. This is usually the first line in the file (line 0). Discovery will disregard the specified line when converting the file to a physical table and creating a logical table.

   If the file has no heading lines, uncheck the **Heading Line** box.

4. In **Skip Lines**, enter the line numbers of any lines to be skipped when Discovery converts the file to a physical table and creates a logical table. Specify lines in the following ways:
   - A single number (0)
   - A hyphenated range of rows, inclusive (10-20)
   - Multiple entries separated by commas (0, 5, 10-15, 18, 27-40)

   **Note:** The first line in the file is Line 0.

**Results**

Continue with the next step, [Add Fixed Width Text File: Column Breaks](#).

**Add Fixed Width Text File: Column Breaks**

**Procedure**

1. In the data preview pane, scroll through the file to make sure that the column headers (if any) and the columns are correctly aligned. The data preview includes all extra blank spaces between columns.
   - If any data is misaligned, exit the wizard, correct the physical file, and restart the wizard from the [Import Text Files](#) step.
   - If the data is aligned correctly, or if you want to proceed without making corrections, continue with the next step.
2. At the beginning of each column, click in the ruler at the appropriate location to add a column break. You can drag a column break to move it, or double-click the arrow in the ruler to delete the column break.

Add all column breaks, scrolling right to display all columns.

Note: Do not add a column break in front of the first column or after the last column.
Results

Continue with the next step, Add Fixed Width Text File: Column Properties

Add Fixed Width Text File: Column Properties

Procedure

One by one, click in each column and review the column's properties. The first column is selected by default.
Results

**Add Text Table**

Select a column in the preview pane to edit its properties.

**Column Properties**

- **Column Name**: EMPLOYEE_ID
- **Column Width**: 12
- **Skip Column**: 
- **Truncate Whitespace**: 
- **Null Value Handling**: TextAsIs

Sample File: C:\Supporting Documents\Text Data Files\W_EMPS.TXT

<table>
<thead>
<tr>
<th>EMPLOYEE_ID</th>
<th>SALUTATION</th>
<th>EMPLOYEE_FIRST_NAME</th>
<th>EMPLOYEE_LAST_NAME</th>
<th>SSN</th>
<th>BEI</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>Mrs</td>
<td>Nancy</td>
<td>Davovic</td>
<td>656-39-5787</td>
<td>5/1</td>
</tr>
<tr>
<td>78</td>
<td>Dr</td>
<td>Andrew</td>
<td>Fuller</td>
<td>464-20-4652</td>
<td>8/1</td>
</tr>
<tr>
<td>91</td>
<td>Ms</td>
<td>Laura</td>
<td>Callahan</td>
<td>266-05-6282</td>
<td>3/5</td>
</tr>
<tr>
<td>15</td>
<td>Mr</td>
<td>Johan</td>
<td>Loehr</td>
<td>385-50-2780</td>
<td>1/1</td>
</tr>
<tr>
<td>27</td>
<td>Mr</td>
<td>Dexter</td>
<td>Fey</td>
<td>645-74-1770</td>
<td>1/1</td>
</tr>
<tr>
<td>34</td>
<td>Mr</td>
<td>Brian</td>
<td>Barton</td>
<td>568-15-7102</td>
<td>1/1</td>
</tr>
<tr>
<td>48</td>
<td>Mr</td>
<td>Danny</td>
<td>Fey</td>
<td>620-22-4833</td>
<td>1/1</td>
</tr>
<tr>
<td>84</td>
<td>Mr</td>
<td>Leney</td>
<td>Slaton</td>
<td>863-00-8674</td>
<td>1/1</td>
</tr>
<tr>
<td>85</td>
<td>Mrs</td>
<td>Giselle</td>
<td>Deets</td>
<td>231-54-6541</td>
<td>1/1</td>
</tr>
<tr>
<td>96</td>
<td>Mrs</td>
<td>Alicia</td>
<td>Callier</td>
<td>629-18-1968</td>
<td>1/1</td>
</tr>
</tbody>
</table>

- **Column Name** matches the column header shown in the preview pane. If there is no header line, the name is COLUMN_0. You can edit the Column Name field. The preview pane displays the new column name when you click out of the field.

- **Column Width** is the width of the longest value or the heading in the column, including empty spaces. You can manually change this value to any positive number or 0. The preview pane does not update to reflect the new column width.

**Note**: Extra spaces are not included in the data preview pane; each column displayed here is only as wide as its longest text string.

The final column in the table does not have a column width determined. This screen does not contain all of the rows in the physical file if the file is large. Discovery cannot tell if this column in the full physical table contains longer values than are shown here.

- **Skip Column** determines whether Discovery will skip (not include) this column when converting the file to a physical table and creating a logical table.

- **Truncate Whitespace** determines whether Discovery will remove empty spaces detected in this column when converting the file to a physical table and creating a logical table. If this is not checked, white space is converted to spaces.
Null Value Handling determines how a null value detected in this column is handled when Discovery converts the file to a physical table and creates a logical table.
- **TreatAsNull** converts it to a null value.
- **TreatAsEmpty** converts it to an empty cell (do not use this option with numberstring columns).
- **TreatAsError** causes Discovery to execute the Error Handling action selected on the next screen.

Continue with the next step. **Add Fixed Width Text File: Table Properties**

**Add Fixed Width Text File: Table Properties**

**Procedure**

1. Verify that the table properties are correct.

2. Click **Finish**.
Results

The physical text file is added to the Data Sets window in the Text Files list, and Discovery and creates a logical table. Although it is listed under Text Files, the physical text file has been converted to a table. Discovery will treat it as a table in all further processing and analysis actions.

To add another physical text file to this table, see Combining Multiple Text Files into One Table. The new text file must be in exactly the same format as the first text file.

To import a different physical text file, repeat the Import Text Files procedure.

Combining Multiple Text Files into One Table

About this task

You can combine multiple physical text files into one single table in the staging data source, as long as all of the text files have exactly the same delimited or positional internal structure. The table keeps the name of the first physical text file imported in that format.

For example, a company that saves each week’s orders as a separate text file can create a useful data source by combining a year’s worth of files into a single table. This table can be processed as one entity rather than as 52 single files.

The logical table and the Full Set sample set contains data for all of the imported files in the physical table. You can create additional logical tables and sample sets for this physical table and perform processing on them. Before you do, see Modify a Logical Table.

To combine multiple physical text files into a single table in the staging data source:

Procedure

1. Import a physical text file with the desired formats (data types, column widths, error handling, etc.), as described in Import Text Files.
2. After finishing the physical text file import, right-click on the Files name or icon under the appropriate physical text file and select Add a new file to the File format. A standard file browser window opens.
3. Navigate to the desired physical text file and click Open.

Results

The file is immediately added to the existing table in the staging data source, and also appears under the Files header. Its contents are added to this table's logical table and its Full Set sample set. You can view the contents of the table by opening the Full Set sample set and clicking Preview Data.

To delete files from the combined table, right-click on the file under the Files header and select Delete the Selected File.

A sample Text Formats & Files pane is shown below.
Import CWM Files as Physical or Logical Tables

CWM (Common Warehouse Metamodel) XML files can be imported into a project as physical tables or as logical tables for all projects.

You can also [Import a CWM File as a Target Table Schema](#) in SDD projects.

Import CWM Files as Physical Tables

CWM files can be imported as physical tables. This is useful for importing a group of tables into a data set as one action, rather than searching the schema for the tables.

About this task

When a table is exported in CWM format, all NUMBERSTRING datatypes are converted to NUMERIC, and are then converted to DECIMAL when imported into Discovery Studio. After importing a CWM file, carefully check the Column Analysis results and change DECIMAL datatypes back to VARCHAR if appropriate.

Procedure

1. Using third-party software, export the appropriate tables from the source schema in CWM XML format.
2. In Discovery Studio, [Create a Project](#) If desired, [Name and Classify the Data Set](#)
3. Use the [Create Database Connection](#) procedure to add a connection to the source schema.
4. Right-click on the database connection and select [Import tables from CWM file](#).
5. In the file browser, navigate to the location of the exported CWM file, select it, and click [Open](#).
Discovery creates a physical table, logical table, and sample set for each table in the CWM file. You can now import additional tables into this data set, populate another data set, or begin processing the data.
Import CWM Files as Logical Tables

CWM files can be imported as logical tables. This is useful for defining logical tables in a third-party tool and importing the logical tables into Discovery for processing, rather than defining the logical tables in Discovery after importing the physical tables.

About this task

Existing PF Key relationships in the logical tables are also imported.

Note: When a CWM file is imported as logical tables, Discovery does not automatically import the source physical tables. The following procedure includes instructions for importing the physical tables and manually associating them with each logical table in the CWM file.

Procedure

1. Using third-party software, define the logical tables and export them from the source schema in CWM XML format.
2. In Discovery Studio, Create a Project. If desired, Name and Classify the Data Set.
3. Use the Create Database Connection procedure to add a connection to the source schema containing the physical tables. If the physical tables are in more than one source schema, create a connection to at least one of the schemas.
4. Right-click on Logical Tables and select Import logical tables from CWM file....
5. In the file browser, navigate to the location of the exported CWM file, select it, and click Open.
Discovery creates a logical table and default sample set for each logical table in the CWM file. Note that the **Database Connections & Tables** list is empty, indicating that there are no physical tables in the data set yet.
6. Import all of the physical tables associated with the logical tables using the standard [Import Physical Tables/File Formats] procedure. You may need to create additional connections if the physical tables are in different schemas.

If you have one or more CWM files containing the physical tables that correspond to the logical tables, you can import the files using the [Import CWM Files as Physical Tables] procedure. Create additional connections, if needed.

Discovery creates a new logical table for each physical table that you import. If these new logical tables are not required for your project, delete them now or after completing this procedure.
7. Double-click the first required logical table to open The Logical Table Definition Screen.

8. In the From field, enter the physical table associated with this logical table. If there is more than one physical table, enter the join condition. For more information, see From.

9. In the Logical Columns grid, define each column in the logical table. For more information, see Logical Columns.

10. If necessary, enter a Where or Having clause. If desired, you can also Remove Duplicates, Validate the logical table definition, and Logical Tables Data Preview.

11. Click OK to close the Logical Table Definition screen.

12. Open the remaining required logical tables and repeat steps 8 through 11.

13. If you did not delete the unneeded logical tables in Step 6, delete them now.

Results

You can now import additional tables into this data set, populate another data set, or begin processing the data.

Importing Lookup Tables

Lookup tables are only used in TD projects. Lookup tables can be database tables or text files.

The procedure for importing a lookup table and making it usable with a project is described in “Import a Lookup Table (Import a Data Rule)” on page 558.

Note that the data rule dataset does not have a Data Set Classification.

Note: If you will use Discovery to discover data rules, you do not need to import lookup tables for the project.

Modify Physical Tables in the Data Set

You can modify the contents of a physical database table or text file at any time. In most cases, you need to re-run all subsequent discovery actions after making any changes.

If you do make any changes to the data set or primary sample set, repeat all discovery tasks, starting with Column Analysis. The new data may cause...
Discovery to discover different statistics, PF Keys, data objects, overlaps, target matches, or mapping results.

**Edit a Physical Table’s Properties**

**About this task**

You can edit certain properties of an imported physical table. This can change existing results that reference this table (including Column Analysis, PF Keys, Data Objects, maps, transformations, and so on), so you should re-run discovery after making any changes.

To edit a physical table’s properties, select the table name in the Import Objects list, right-click on it, and select **Edit the selected Table/File Format**. (You can also double-click the table name.)

The physical table display name cannot be changed. However, you can change the full name of the physical table.

The **Column name prefix** and **Number of Columns** are displayed here, but are configured in **The Logical Table Definition Screen**.
Edit a Text File’s Properties

About this task

You can edit some of a text file’s properties that were set during import. The text file’s corresponding logical table and default sample set are updated immediately.

To edit a text file’s properties:

Procedure

1. Select the text file name in the Import Objects list, right-click on it, and select Edit the selected Table/File Format. You can also double-click the file name.

2. Edit the properties as necessary. The Table Name and Column Name fields cannot be changed.
3. Click OK to save the changes.

**Add a Physical Table or Text File**

You can [Import Database Tables](#) or [Import Text Files](#) to the data sets at any time in the mapping process. You can also reload or reimport data, create new logical tables, define and select a new primary sample set, and modify the physical tables or text files after you have imported them.

Re-run all discovery tasks after adding a table or text file.

**Point to a Different Text File**

**About this task**

You can change the specified location of a text file in the data set at any time. This is useful when a modified or appended version of the text file is available and the file has been renamed or moved to a different directory.

Re-run all discovery tasks after changing the location of a text file.
Note: Do not use this procedure if the changed text file contains additional columns, or if any column data types have changed. In that case, delete the previous text file from the data set, add the modified one as a new file, and re-run all processing steps.

Procedure

1. In the Data Sets window, right-click on the text file name and select Edit the selected Table/File Format.

2. In the Text File Properties window, click Browse.
3. Navigate to the updated file and click **Open**.
4. In the **Text File Properties** window, click **OK** to close the window.
5. Re-run all subsequent processing steps.

**Delete a Physical Table or Text File**

**About this task**

Select the table or text file name, right-click on it, and then select **Delete the selected Table/File Format**. This removes the table from the Discovery staging data source but does not affect the original database table or text file. Depending on how far you have progressed in the project, you may need to re-run some or all discovery tasks completed so far.

Deleting a physical table or text file after you have begun processing the data has an effect on all discovery results that use the table or file. For example, in all projects, PF Keys and Data Objects that involve the table or file become invalid.

TD and SDD projects are affected as follows:

- In TD projects, all map expressions involving the table or file become invalid, and the map itself may become invalid if the table or file is used in the primary join condition, primary binding condition, or primary where clause.
- In SDD projects, the Overlaps statistics can become invalid, and the target tables and Unified Schema PF Keys may also become invalid.
When you delete a physical table or text file in any project, re-run all processing steps to obtain new results.

If you delete a physical table or text file and then reimport a physical table or text file with the same name, and create a new logical table with the same name, re-run all steps with Only Refresh Statistics checked. This revalidates all results.

**Keeping Dependent Logical Tables:**

**About this task**

Deleting a physical table or text file deletes all logical tables dependent on that physical table, even if the logical table also depends on other physical tables. For example, assume that LogicalTable1 depends on PhysicalTableA and TextFileA. If you delete PhysicalTableA, then LogicalTable1 is also deleted.

To keep a dependent logical table when you delete one of its physical tables or text files, modify the logical table before deleting the physical table or text file so that the logical table does not depend on the table or file. You can then safely delete the physical table or text file.

**Delete a Logical Table**

You can delete any logical table, including the default logical table created for a physical table or text file. Discovery processing is only performed on logical tables, so if you delete the only logical table containing the data from a physical table or text file and then begin processing the project, the data in that physical table or text file will not be included in any processing.

Deleting a logical table after you have begun processing the data has an effect on all discovery results that use the logical table. For example, in all projects, PF Keys and Data Objects that involve the logical table become invalid.

**Note:** You cannot delete a logical table if any of its logical columns are used in a Unified Schema Builder target table. To delete the logical table, first remove the logical columns from the target table.

TD and SDD projects are affected as follows:

- In TD projects, all map expressions involving the logical table become invalid, and the map itself may become invalid if the table is used in the primary join condition, primary binding condition, or primary where clause.
- In SDD projects, the Overlaps statistics can become invalid, and the target tables and Unified Schema PF Keys may also become invalid.

When you delete a logical table in any project, re-run all processing steps to obtain new results.

If you delete a logical table and then recreate a logical table with the same name, re-run all steps with Only Refresh Statistics checked. This revalidates all results.

**Reload a Logical Table**

Reload a logical table when more rows have been added to a physical table or text file used in the logical table, but the metadata or number of columns has not changed.

Select the logical table in the Logical Tables section, right-click on it, and then select Reload.
Discovery drops the staging table(s) from the Staging Data Source and recreates them automatically the next time they are needed. Any custom sample sets are also updated.

Re-run all discovery tasks after reloading logical tables.

**Reimport a Physical Table**

Reimporting a physical database table is necessary when the metadata or the number of columns in the physical table have changed.

**About this task**

**Note:** Text files are not reimported. See [Update Text Files](#).

Select the physical database table in the Import Objects Pane, right-click on it, and then select Reimport. You can select multiple data sources within the same data set using SHIFT-click or CTRL-click. You can also select multiple physical tables within the same data source using SHIFT-click or CTRL-click.

Discovery drops the staging table(s) from the Staging Data Source and recreates them automatically the next time they are needed.

Because Discovery does not know how the new version of the physical table differs from the old version, all PF Keys for the old table(s) are dropped when a table is reimported. This includes defined PF Keys as well as discovered PF Keys.

After reimporting a table, [Validate](#) the project to identify any invalid expressions or objects. For example, if a column has been renamed, data rules that reference that column name are not automatically updated. Manually update invalid expressions or objects, or re-run discovery in those steps to correct the validation errors.

If the new physical table contains any PF Key metadata, that information is used to create new PF Keys.

**Update Text Files**

**About this task**

To update a text file with a version containing additional or fewer columns, delete the physical file and its associated logical tables using the [Delete a Physical Table](#) or Text File procedure, and then add the new version as a new physical file using [Import Text Files](#).

Re-run all discovery tasks after this procedure.

**Add or Modify Sample Sets**

To add or modify a sample set, see [About Sample Sets](#).

**Add or Modify Logical Tables**

To add or modify a logical table, see [About Logical Tables](#).
Select a Primary Key Column

About this task

This action is only necessary for certain logical tables in TD projects that will be validated using IBM InfoSphere Exception Manager. Not all logical tables require that you select a primary key column. For more information, see Create Validation Jobs.

A primary key column does not need to be in a primary-foreign key relationship. Any column that is unique and whose existing values will not change can be selected as a primary key column. You can also select a combination of columns to use as the primary key for that logical table.

Note: If you cannot currently identify a primary key column in the logical table, continue with normal discovery steps until you Discover PF Keys. Use the PF Key discovery results and Review Column Analysis Data as necessary to identify an appropriate primary key column, then return to the Data Sets tab and select the primary key column using the following procedure.

Procedure

1. In the Data Sets tab, right-click on the logical table name (not on a sample set) and select Set Primary Key Columns.

2. In the Primary Key Columns dialog, highlight the primary key column and click > to move it from the Columns in the Table list to the Primary Key Columns list.
If you move more than one column into the **Primary Key Columns** list, Exception Manager considers the primary key as a combination of those columns, in any order. For example, if you select EMPLOYEE_ID and SSN as primary key columns, Exception Manager recognizes both <EMPLOYEE_ID><SSN> or <SSN><EMPLOYEE_ID> as valid primary key values for this logical table.

3. When you have selected the primary key column(s), click OK to close the **Primary Key Columns** dialog.

**Results**

The primary key column is now selected for that logical table.

**About Logical Tables**

A logical table is a virtual table within Discovery. It is a mapping of columns in one or more physical tables or text files in a data set.

Logical tables allow you to normalize data, combine two or more columns into one column, combine data from different tables, reduce data duplication among data sets, drop irrelevant columns, standardize business terms, and customize your data in other ways before processing.

Logical tables provide faster and more accurate results because data analysis is focused on tables containing exactly the information needed for the business project. The data can be tailored more closely than with sample sets alone.
All processing within Discovery is performed on logical tables. In addition, all sample sets are created on logical tables.

**Overview**

When you import a physical table or text file into a data set, Discovery automatically appends the physical table with _PT (for physical table). Discovery then creates a logical table that uses the original table or file name. This logical table contains all of the data from the original physical table or text file. In addition, Discovery creates a default full set sample set from the logical table, also containing all of the data from the physical table.

**Note:** All processing is performed on logical tables. The physical tables and text files with the _PT suffix appear only in the Data Sets tab. All other screens in Discovery display logical tables.

You can define additional logical tables in the project. Discovery treats each logical table as a separate object in the data set. For example, if you import three physical tables into a data set, the data set contains three logical tables. If you then create another logical table using one of the physical tables, Discovery considers that the data set now contains four tables. When you run analysis tasks, results are reported for each one of the four logical tables.

Discovery performs all analysis and processing tasks on each logical table’s primary sample set. You can create additional sample sets for each logical table and change the primary sample set as needed.

**Restrictions on Logical Table and Logical Column Names**

The following restrictions apply to logical table names and to logical column names.

- Cannot contain **Reserved Keywords**
- Must start with an alphabetic character or _ (underscore)
- Cannot contain a wildcard character (other than _ as the first character)
- Cannot include quotation marks
- Must be 30 characters or less

In addition, logical tables used in maps (TD projects only) cannot be renamed.

**Maximum Number of Logical Columns**

Projects can contain a maximum of 50,000 logical columns in any number of logical tables. The total number of physical columns can exceed this amount, but only if the logical tables have been modified to contain less than 50,000 columns.

If the import of a physical table will result in more than 50,000 logical columns in a project, an error message will appear during the physical table import.

To reduce the number of logical columns in a project, modify the existing logical tables to remove unnecessary logical columns and then re-import the physical tables.

**Define a Logical Table**

One logical table is automatically created for each physical table or text file when it is imported into the project.
About this task

You can define additional logical tables using one or more imported physical tables or text files at any time during the project. Before you do, see [Modify a Logical Table](#).

After defining a logical table, re-run all subsequent processing steps.

To define a new logical table:

Procedure

1. Import Database Tables or Import Text Files into a data set.
2. In the Data Sets tab, right-click on a physical table or text file format.
3. Select the appropriate option:
   - **Add a new logical table** creates a new logical table containing the exact same data as the physical table.
   - **Add a new logical table with processing** opens the Import Process Options window, which allows you to specify SQL commands to perform on columns of String, Numeric, or Date Type data types in the physical table. The results are saved as the new logical table. Enter the appropriate SQL for one or more of the listed data types and click OK. (You can perform these commands on all logical tables using the Tools>Options> Import Processing option.)

For example, entering the notation `upper($c)` in the String type column field creates a logical table with the `upper` function applied to all string type
columns.

4. In the **Logical Table Definition Screen**, define the logical table as appropriate. See [Restrictions on Logical Table and Logical Column Names](#).

5. When you are finished, click **OK**. The new logical table is listed in the **Logical Tables** section of the **Import Objects** pane.

   Preview the logical table data on the Column Analysis page and in the **Logical Table Definition Screen**. You can also preview the logical table's sample set on the **Sample Set Definition Screen**.

### Delete a Logical Table

You can delete any logical table. Deleting a logical table also deletes all of the logical table's sample sets. If you delete the only logical table representing a physical table, that physical table will not be used in any processing.

#### About this task

You can delete a logical table.

**Note:** Deleted logical tables cannot be recovered.

In the **Logical Tables** section of the **Import Objects** pane, right-click on a logical table and select **Delete logical table**. The logical table and all of its sample sets are immediately deleted.
Modify a Logical Table

You can modify any logical table.

About this task

In some cases, modifying the default logical table or defining new logical tables results in output that cannot be used by a third-party application.

For example, the application may have a restriction involving modified logical tables. If any logical table in the map contains less than the full content of the corresponding physical table, the application cannot use that exported XML to run an ETL job.

If you are planning to use project output in a third-party application, avoid this problem by making sure that modifying the logical tables will not affect the application's use of the exported data.

Note: After modifying a logical table, re-run all subsequent processing steps. If the logical table uses SQL functions or procedures that are only available in the staging data source, you should explicitly re-stage the table with Re-run Column Analysis before re-running the subsequent processing steps.

Procedure

1. In the Logical Tables section of the Import Objects pane, right-click on a logical table and select Edit logical table.
2. In [The Logical Table Definition Screen](#) make the necessary changes. See [Restrictions on Logical Table and Logical Column Names](#).

3. Click OK. The logical table is updated.

4. Re-run all subsequent processing steps.

### Rename a Logical Table

**About this task**

You can rename logical tables. However, in TD projects, do not rename logical tables after you have discovered maps because the map expressions will become invalid. See [Restrictions on Logical Table and Logical Column Names](#).

To rename a logical table:

**Procedure**

1. In the Logical Tables section of the Import Objects pane, right-click on a logical table and select Rename logical table.

2. In the popup, change the name.

3. Click OK. The name is immediately updated in all Discovery screens.

4. If you have already performed some processing, explicitly re-stage the logical table using the [Re-run Column Analysis](#) task and selecting Only Stage Selected Items.
The Logical Table Definition Screen
The Logical Table Definition screen is shown below. This screen allows you to define and modify logical tables.

About this task
The fields and panes on the screen are described below. Modify each field as necessary, then click OK to save the changes and close the screen.

The Logical Table Definition screen has the following fields and buttons:
- Name
- Notes
- Column Name Prefix
- From
- Logical Columns
- Where
- Group By
Name
By default, the logical table is given the name of the original physical table (without the _PT). Subsequent logical tables are numbered consecutively, as in <logical_table>_1, <logical_table>_2, and so on. You can change this name to be more descriptive of the actual contents.

Notes
Allows you to enter comments about this logical table

Column Name Prefix
Column names often have a prefix that identifies the table they belong to, the database itself, or some other information. Because these are the same for all columns in a table, they are not useful for analysis and Discovery ignores them. If the column name prefix is incorrect, change or delete it here.

From
About this task
By default, this field shows the physical table that the logical table is based on. If you are adding columns from one or more physical tables in the data source, enter the other physical table name(s) here with a join condition. You can also drag the physical tables from the Physical Tables pane into this field.

Start typing in the field to display the Expression Editor

For example:
HQ_EMP_PT join W_EMPS_PT on HQ_EMP_PT.NAME2 = W_EMPS_PT.EMPLOYEE_LAST_NAME

When defining the join condition, note the following characteristics:
- A logical table can refer to one or more physical database tables and text files.
- The physical tables in a logical table can have different database connections and be of different database types.
- The same physical table can be referred to by two or more logical tables.

Logical tables have the following limitations:
- Logical tables cannot refer to sample sets or other logical tables.
- A logical table cannot have more than one reference to the same physical table or text file (self-join). However, you can import the physical table or file into the data set several times and have a reference to each one.
- Aliases are not supported in the join condition.
Logical Columns

About this task

All of the columns that will be in the logical table are listed here. By default, the list contains all of the columns in the physical table that the logical table is based on. See Restrictions on Logical Table and Logical Column Names.

There are several ways to add physical columns to this logical table:

- Click + and enter the new column name and expression. The Data Type, Length, Precision, and Scale are automatically filled in.
- Drag one or more columns from the Physical Tables pane into the Logical Columns list.
- Drag a table into the Logical Columns list to add all columns in the table.

Note: The new column's parent table must be in the From statement.

To delete a column from the logical table, select the row and click \( \times \). The column will not be included in the logical table, but it is still present in the physical table.

The columns in this grid are:

- Ordinal
- Column Name
- Column Expression
- Data Type
- Length
- Precision
- Scale

Where

A logical table can have only one Where expression. The expression is based on the physical tables and columns.

About this task

Start typing in the Where field to display The Expression Editor. Place the filter value in single quotes, but do not include the actual string WHERE; however, you can use ANSI SQL functions like substr, cast, and the concatenation operator ||.

Group By

You can create a group by expression with any physical table column used in the logical table. The logical column expression must contain an aggregate function.

Having

If a group by expression exists, you can create a having clause.

Remove Duplicates

When this checkbox is selected, Discovery will ignore (not include) any duplicate records it detects in the logical table. The rows must be exact duplicates with all values in the row matching exactly. Only one row with those values will remain in the logical table.
Validate

Click this button to validate the logical table definition. Discovery Studio records up to the first 500 errors. The error log is cleared at the beginning of each validation. See Error List and Output Pane.

Row Count, Update Count

Click this button to update the Row Count field with the number of rows in the logical table, given the current definition.

Logical Tables Data Preview

Click Preview Data to display the contents of the logical table, given the current definition. You can sort columns in this preview. Use Preview Criteria to customize the data preview.

The following grid icons are available:
- Advanced Search
- Search
- Auto Filter Row
- Column Chooser
- Export Grid Content

About Sample Sets

This section describes how to create, modify, and use sample sets in a Discovery project.

Overview

Discovery can ignore some of the data in your project and analyze only the data you select. For example, you may want to focus on customers who live in a certain geographic area, or on accounts that contain more than a certain number of transactions after a specific date. With very large data sets, you can save processing time by analyzing a representative subset of the data and then applying the results to the entire data set.

Creating sample sets allows you to select specific data for analysis in the processing and discovery tasks. Discovery creates one sample set per table by default, containing all of the data in that table. You can create additional sample sets for each table, containing as much or as little data as required, and select which sample set for each table will be used for analysis. All data not included in the selected sample set is ignored.

In order for the results to be correct, it is very important that the sample set be representative of the data in the full data set.

What is a Sample Set?

A sample set is a copy of a table or text file that contains some or all of the records in that table or file. Sample sets can only be created on tables and files that have been imported into a Discovery project. Sample sets are stored in the Discovery repository. Creating, modifying, or deleting a sample set does not affect the original table or text file.
The contents of a sample set are defined by SQL select statements, where clauses, and Order By statements. A sample set can also be defined by a join statement on two or more tables. Each table or text file in a project can have as many sample sets as needed for the purposes of the project. Each table or file must have one sample set selected as the primary sample set used for processing tasks. You can select a different primary sample set at any time. However, you usually need to re-run some or all tasks after changing any table’s primary sample set.

Sample sets speed up analysis and discovery tasks on large tables by reducing the number of records that are processed in each step. For example, assume a TD project contains very large source and target data sets. You create smaller sample sets that contain representative data for each table, make them the primary sample sets, and then perform transformation discovery and analysis. After validating and refining the results, you return to the Data Sets screen, change each table’s primary sample set back to the Full Set data set, and then refresh the statistics in the Transformations tab. If the representative sample sets were correctly specified, the transformations will be valid for the full data set. If the results are not acceptable, you can refine the transformations or create new sample sets containing more accurate samples, then repeat discovery.

When you import a table or text file into a project, Discovery automatically creates a default sample set for the table or file and selects it as the primary. This default sample set is named Full Set and contains the full contents of the table or file. Each table’s Full Set sample set is used for all processing and discovery tasks unless you create another sample set and select it as the primary.

All of the sample sets for a table or file are listed in the Data Sets screen under the entry for that table or file. Each current primary sample set is indicated by a green dot.
In order for a sample set to be useful, it must be representative of all data in the table. You can only determine that a sample set represents an entire table if you are familiar with the data in that table.

For example, suppose you have a large table of historical sales data that is partitioned by month, and the structure does not change from one month to another. You can create a sample set containing the data from only one month and running the analysis and discovery tasks on that sample set, then apply the resulting transformations to the entire table.

However, you could not create a sample set this way for a table that is not reliably partitioned by time with consistent structure. Other factors may also prevent a useful sample set from being created from a table. In some instances, there is no alternative to using the Full Set.

**The Primary Sample Set**

Each table always has one primary sample set. By default, this is the Full Set sample set. After you create additional sample sets, you can select any sample set as the primary.
The Full Set Sample Set

Discovery automatically creates one sample set, named Full Set, for each table or text file in the data set. The Full Set sample set contains all of the data in the table or text file and cannot be modified, renamed, or deleted.

When To Use Sample Sets

Only use a sample set when you are familiar with the source and target data, or when you know which tables will benefit from sample sets. If you know the data, you can define sample sets before you do profiling. Then, based on the profiled sample set, you can discover data objects, define maps, and so on. After verifying the discovery, apply Sample Set Results to the Full Data Set.

Sample sets are useful to map horizontally partitioned target tables. It also generally saves time to perform discovery on large data sets.

Creating and Deleting Sample Sets

About this task

At any point in processing, you can create a new sample set, revise an existing sample set, or designate a different sample set as primary.

- You can create any number of additional sample sets containing filtered table data or selected rows.
- You can manually create a valid sample set in the Data Sets window for data that you are familiar with. In this case, you would use the Limit Results To option to set the maximum number of rows, create a Select or Where statement, and preview the data. Continue modifying the filters until the sample set accurately represents the data in the entire table.
- You can automatically create a Related Sample Set in the Data Objects window.

Note: If you modify the primary sample set or designate a new one as primary, re-run all subsequent processing steps to make sure the discovery results are still accurate.

You can delete any sample set you have created. The following sample sets cannot be deleted:

- the Full Set sample set
- the current primary sample set

Determining if a Sample Set is Valid

A sample set must be as representative as possible of the full table. This consideration is key to determining the filter criteria you use to define a sample set.

For instance, if you filter an EMPLOYEES table for STATUS=’temp’, the resulting set probably won’t represent the whole table in significant ways, which may lead to incorrect results. Similarly, if you take only the first N rows in a table but you are not familiar with the table’s content, the sample again may not be representative.

Also, when mapping one sample set to another, avoid using sample sets that “miss” each other by failing to contain relationships that are present in the full sets. For example, suppose you are mapping two employee tables. One table’s sample set is filtered by LOCATION=’midwest’, and the other table’s sample set is
filtered by GROUP='marketing'. Each sample set is valid and provides a representative sample of the full set, but if all of the company’s marketing employees work in the New York office, the sample sets will erroneously indicate that the tables have no relationship.

**Define a Sample Set**

By default, Discovery creates a sample set named Full Set for each database table or text file you import into a data source. This is the default primary sample set and contains all data in the imported table or file.

**About this task**

For each database table or text file, you can define additional sample sets that contain the full data set or only a subset of the data. When you make the new sample set the primary, Discovery uses the data in that sample set for all discovery and analysis tasks involving that table or file.

**Note:** If you modify the primary sample set or designate a new one as primary, re-run all subsequent processing steps to make sure the discovery results are still accurate.

A database table or text file must always have at least one sample set, and only one sample set at a time can be the primary. The sample set marked as primary cannot be deleted. When there is only one sample set, it is the primary by default and cannot be deleted.

**Note:** Whenever you create or define a sample set, always click **Validate** to check for syntax or other errors.

To define a new sample set for a database table or text file:

**Procedure**

1. Display the Data Sets window.
2. Right-click on the table or file name and select **Add a new Sample Set**. The **Sample Set Definition** screen appears.
   - The Sample Set Definition screen displays the fields in a Select statement: **Select, From, Where, and Order By**. It also allows you to limit the number of rows in the sample set, remove duplicates, identify how many rows will be in the sample set, and preview the data.
3. In the **Name** field, enter a name for the new sample set.
Changing the Select Statement Procedure

If the default Select statement of all columns in the table is correct, skip this step and proceed to Creating a From Expression. If the Select statement needs to change, click Edit Column Expression. The resulting screen shows all columns in the selected table along with the data type, precision, length, and scale of each column.
1. To mask the values in a column, replace the default text (TABLE_NAME.COLUMN_NAME) in the Edit Expression field with the string or expression that you want to use as a mask.

   - A string must be enclosed in single quotes. For example, to mask the values in the column HQ_SALARIES.EMPLOYEE_ID with asterisks, replace the value HQ_SALARIES.EMPLOYEE_ID with the string '********'. If the string exceeds the column width, it will be truncated.

   - An expression must be compatible with the column type. If the resulting string exceeds the maximum column length, it will be truncated. For example, to display only the first character of the FULL_OR_PART_TIME column, enter `substr(HQ_SALARIES.FULL_OR_PART_TIME,1,1)`.

2. Click OK. The modified expression appears in red text in the Select field, replacing the column name.

3. Click Preview Data to refresh the data display. The string or expression appears in the specified columns. Actual data appears in all other columns.

Note: You can restore the original values at any time by replacing the modified expression in the sample set with the original TABLE_NAME.COLUMN_NAME value.
Creating a From Expression

Procedure

1. If this sample set is not related to any other table's sample set, skip this section and proceed to Creating a Where Expression.

   Click to add an empty line in the From field.

2. Left-click Choose Table to display a drop-down list of other tables in the data source.

3. From the list, select the table containing the desired sample set.

4. Left-click the entry in the Sample Set column and select a sample set. All existing sample sets for the table you just selected are listed.

5. In the Join Type column, either leave the default condition as Join or left-click on it to select Left Outer Join.

6. Click in the empty row under Join Condition to display The Expression Editor.
7. Type or paste the join expression that represents the relationship between these two sample sets. This is the same expression displayed in the PF Keys Window when you select the join in the diagram or in the Connected Tables list.

**Note:** When you create sample sets through a data object, related samples sets are discovered automatically. See [Create a Related Sample Set](#). You can relate additional sample sets and can change the order of the additional sample sets in the list by highlighting them and clicking the up and down arrow buttons.

### Creating a Where Expression

**Procedure**

If the sample set does not need a Where expression, skip this section and proceed to "Creating an Order By Expression".

Create a Where expression in The Expression Editor. The sample set can have only one Where expression. Place the filter value in single quotes, but do not include the actual string `WHERE`; however, you can use standard SQL functions like `substr`, `cast`, and the concatenation operator `||`. You can [Define Project Variables](#) for use in the Where expression of any sample set in the project. These same variables are available for pre-execution and post-execution scripts when you [Create a Validation Job](#). Use standard JavaScript restriction in variables. Variables must be alphanumeric and start with a non-numeric character.

### Creating an Order By Expression

**Procedure**

1. If the sample set does not need an Order By expression, skip this section and proceed to [Limit Results and RemoveDuplicates](#).

   Click + to add an empty expression to the **Table** column.
2. Click in the **Table** column and select a table name from the list.

3. Click in the **Column** column and select a column name. The list contains all columns in the selected table.

4. Click in the **Sort Type** column and select either ASC for ascending, or DESC for descending sort order.

5. Create additional sort order entries as needed. Entries can be moved up and down in the list using the up and down arrow buttons.

**Limit Results and Remove Duplicates Procedure**

1. To limit the number of rows in your set, check the Limit results to box and enter a value in the rows field. The sample set will not exceed this number of rows.
2. To have Discovery ignore (not include) any duplicate records it detects during sample set generation, check Remove duplicates. If any rows are exact duplicates (all values in the row match exactly), the duplicate rows are removed, leaving only one row with those values in the sample set.

**Preview Sample Set Contents**

**Procedure**

1. To limit the data that appears in the Preview pane, click Load Criteria. (Entries in this screen do not affect the sample set definition, just the display in the Preview pane.) Click here to see Common Grid Icons in Data Preview Screens.
   - To limit the number of rows that appear, check Limit results to: and specify the maximum number of rows to display.
   - If desired, create a Where expression for the preview in The Expression Editor.

2. Click Preview Data to display a preview of the sample set data in the Preview pane, given the current sample set definition.

**Results**

The standard tools for Grid Panes are available in the sample set data preview.

**Modify a Sample Set**

**About this task**

You can modify the sample sets, if desired. Under each table entry is a sample set containing the full contents of the table. For more information and instructions, see About Sample Sets.

- To edit a sample set (including changing the name), right-click on it and select Edit the selected Sample Set. Make the changes, then click OK.
To add a new sample set to a table, right-click on the table name or on any sample set listed under it, and select **Add a new Sample Set**. Select the data to be included in the new sample set, name the sample set, and then click **OK**.

To select a new primary sample set, click its radio button.

**Note**: If you modify the primary sample set or designate a new one as primary, re-run all subsequent processing steps to make sure the discovery results are still accurate.

**Manage Sample Sets**

Discovery allows you to examine, modify, and synchronize groups of sample sets. Sample set grouping can be based on column name (such as primary key or foreign key columns) or on logical table names. The groups can be within a data set or include all data sets.

In the Data Sets tab, click **Manage Sample Sets**. Select either **All Data Sets** or a particular data set, and then select the appropriate view or action:

- The **Manage Sample Sets, Column Groups View** displays the name of each column group along with the number of tables that it is used in. This view allows you to synchronize sample set based on column properties. Because tables containing similar column are often sampled the same way, this view allows you to re-synch (update) a collection of related sample sets. The lower part of the screen displays details about what has been selected in the top grid.
- The **Manage Sample Sets, Logical Tables View** displays each logical table in the data set or in the project, along with the number of columns in each logical table. The lower part of the screen displays the names and other information about the sample sets in the selected logical table.

You can **Sync Sample Sets**. This allows you to apply certain parameters to several sample sets at the same time, including filters, maximum number of rows, and other information.

**Manage Sample Sets, Logical Tables View**

The Logical Tables View displays all of the logical tables in the project, the number of columns in each logical table, and other information. The lower part of the screen contains information about the selected logical table, including whether the logical table's sample set is the primary sample set.

**Manage Sample Sets, Column Groups View**

The Column Groups View displays all the columns in a set of tables (either from a data set or project), in groups based on matching Column Name, Native Type, and Discovered Type. It also displays metadata about each column. When you select one or more related groups of columns, the details for each column in the union of these groups is displayed in the lower grid. You can now create, verify, or modify the sample sets of the tables for these related columns.

**Sync Sample Sets**

The Sync Sample Sets window allows you to create individual sample sets for multiple tables in one action, using a single where clause that automatically contains the appropriate table and column name. Each sample set can then be manipulated individually, or you can update all sample sets that use that specific where clause. This is a useful preliminary step in very large data sets where primary-foreign keys are not known.
About this task

For example, assume you are working on a project containing 10,000 tables with many containing over a million rows, with few known relationships. Manually creating sample sets for Discovery to identify relationships would be extremely difficult and prone to error. In a situation like this, you would create the data sets and run Column Analysis, and then use Manage Sample Sets to view the column groups. This allows you to identify potential related columns in the data set or in the project. For example, if a column group contained 100 columns named CUST_ID, you could select a group of these columns and view their MIN, MAX, and Cardinality values in the Details pane to determine if they contain similar data.

Upon further review of the Column Groups list, assume you find additional column groups with names like CUSTOMER, CLIENT_NO, and C_ID, containing a total of another 150 columns. The details for those columns groups reveal a similar MIN, MAX, and Cardinality, so you decide to create sample sets for these columns and run discovery to find potential relationships. Given the fact that these columns are in 250 very large tables, you would need to create 250 individual sample sets, one for each table, with the exact same where clause. (Comparing one table’s customer IDs of 1-1000 against another table’s customer IDs of 5000-6000 is not likely to result in discovered relationships.)

Sync Sample Sets allows you to define, name, and create these 250 individual matching sample sets in one action. For example, based on the MIN and MAX ranges, you may choose the name CUSTOMER_ID and define a where clause that selects only customer IDs between 1000 and 2500. This action would create 250 new sample sets named CUSTOMER_ID, one for each table with a customer ID column. Each sample set uses exactly the same SQL but contains the appropriate table and column name.

At this point, you can update the sample set row counts to identify how many records are in each sample set. Depending on the results, you can refine the where clause or apply it to additional column groups. When the row count is sufficient, you can use Sync Sample Sets to make each sample set the primary and then run discovery to look for relationships.

Note: Make sure that all selected column groups have a compatible data type.

Procedure

1. In the Manage Sample Sets window, select View: Column Groups.
2. Identify one or more column groups that are likely to contain overlapping data. You can identify columns based on column names and data types, or highlight one or more column groups and look in the Details pane (using View: Columns) to compare their cardinality, selectivity, Min, Max, or other statistics.
3. Select the desired column groups in the upper grid. The individual columns in the column groups are shown in the Details pane.
5. In the Sample Set Properties window, change one or more of the properties.
   • Where: Click in the Where Value field and then click in the white SQL editor below the rows. Begin the where clause with the word Column with a space after, and then enter the SQL.
   • Primary contains a dropdown menu allowing you to make these sample sets primary or not primary.
Limit sets a maximum row count for each sample set.
Name is the new sample set name.

6. Click OK to create the sample sets. Each selected table now contains a new sample set with the specified name, maximum row count, and where clause. If you selected Primary, each new sample set is now the primary sample set for the table.

For the following steps, work in the Details pane using View: Sample Set. Sort the list by the Name column to sort by sample set. A table can have more than one sample set, so tables may appear several times.

7. To view the number of records in one or more sample sets, select the desired sample set for each table and click Update Sample Set Row Counts, in the same dropdown menu as Sync Sample Sets. The Row Count column is updated with the number of records in each selected sample set.

8. To modify one or more sample sets, select the desired sample set for each table and click Sync Sample Sets. Change any properties as described above, then click OK. The selected sample sets have the new properties.

Note: You can rename sample sets in this window, but cannot delete them here.

Map Horizontally Partitioned Target Tables

In some cases, a single target table actually consists of several partitions, each generated from different source tables and using different transformations. Sample sets can be useful here.

About this task

For example, a customer target table may contain customers from SAP, PeopleSoft, and OracleApplications. If the SAP source is mapped against the entire target table, the hit rate is low because non-SAP customers are shown as misses. Also, binding conditions may not be discovered.

To map horizontally partitioned target tables using sample sets (using SAP, PeopleSoft, and OracleApplications tables as examples).

Procedure

1. Create a sample set for the target table that contains only customers from SAP.
2. Using the SAP sample set, map the SAP source tables to the target table. Refine this map until it is good.
3. Create another sample set for the target table that contains only PeopleSoft customers.
4. Using the PeopleSoft sample set, map the PeopleSoft source tables to the target table, and refine this map also.
5. Create an OracleApplications sample set for the target table.
6. Using the OracleApplications sample set, map the OracleApplications source tables to the target table, and refine this map.

Results

In this scenario, three maps were discovered, one for each customer source. Sample sets allowed Discovery to focus on the right data for discovery.
Perform Discovery on Large Data Sets

When a data set is very large, it can be time-consuming to perform correlation and transformation discovery on the full data set. Generally you don’t want to use a very large data set for discovery unless there is no other option. This procedure describes how to execute the discovery tasks on small sample sets and then apply the discovered results to the entire data set. There can be two outcomes to this activity:

- The transformations discovered using the sample sets are correct, and the statistics that result when you apply the results to the entire data set are satisfactory.
- The transformations discovered in the sample sets do not yield acceptable statistics when you apply the results to the entire data set are not satisfactory.

The first outcome is, of course, the desired one, and occurs when the sample set accurately represents the contents of the large data set. When the discovered results are not acceptable, it is usually because the data in the sample set was skewed. These results can be improved by refining the transformations using data that is not skewed.

Overview of Procedure:

Transformation discovery is the most processing-intensive step. The other discovery steps involve less analysis and complete more quickly in a large data set. Briefly, the steps involved in this procedure are:

**Procedure**

1. On the full data set, perform all steps up to and including discovering binding conditions.
2. Create sample sets.
3. Perform transformation discovery on the sample sets.
4. Review the results.

**Results**

These steps are described in detail in the following section.

Perform Discovery Using Sample Sets:

**About this task**

You can create a sample set to reduce processing time for a very large source or target (or both) data set.

**Procedure**

1. Perform Column Analysis using the full data sets.
2. Discover PF Keys and data objects.
3. If you do not know which source tables map to which target tables, Discover Target Matches using the full data sets.
4. Display the Data Sets step.
5. Define a Sample Set from the large tables used in the map: the source table(s), the target table(s), or all tables in the data sets. Use one of the methods described below, and use recognizable names, like RSS_Target_<table> and RSS_Source_<table> (where RSS stands for representative sample set).
   - Use a shared column.
If there is a binding condition, create a small sample set of several thousand rows by restricting the binding condition on the full source and target data sets. For example, if the binding condition is `Source.SBC1 = Target.TBC1`, you can create a sample set from the source data set where `SBC1 like '1000%'` and another one from the target data set where `TBC1 like '1000%'`.

6. Make the new sample sets the primary sample sets.
7. Define a Map using the new primary data sets and then Generate Binding Conditions.
8. Display the Transformations tab and Re-run Transformations to discover transformations.
9. Review Transformations results to identify any columns that are not representative of the full data set, based on the transformation expression and cardinality.
   a. In the Transformations tab, identify the target columns that map to a constant value.
   b. For each of those target columns, check the Column Analysis tab to see if its cardinality is greater than 1. (These cardinalities are taken from the full table.)
   c. Determine the next step, based on the results.
      • If no target columns map to a constant, or if none of the target columns that map to a constant have a cardinality greater than 1, the sample set is likely to be representative of the full data set. Skip to Step 13.
      • If any target columns map to a constant and have a cardinality greater than 1, the sample set is not representative of the full data set; a target column with cardinality >1 contains multiple values and cannot have a constant transformation. When this result happens, the sample set contains only one of the multiple values in the column. Continue with Step 10.
10. Return to the Data Sets step and create a new sample set for the target table, using the following procedure:
   a. For each target column `CC_i` that maps to a constant and has cardinality >1, given a Mode value `M_i`, add a disjunction of `CC_i != M_i`. The goal is a where clause for generating samples. For example:
      `CC_1 != M_1 or CC_2 != M_2 or ... CC_n != M_n`
   b. Apply each where clause to the target table's full sample set.
      This creates a new target sample set containing all of the non-mode values in the columns from Step 9. Because all of the predicates are OR, the mode will likely be included as well. For example, rows where `CC_1` is not `M_1` will most likely have `CC_2` of `M_2` and `CC_n` of `M_n`.
      If the new target sample set is very small (because all anomalies occur in a few rows), you can combine it with the target sample set created in Step 5 to add more mode values. For example, add or `Target.BC1 like '1000%'`.
11. Make the new target sample set(s) the primary.
13. Return to the Data Sets step and, for the source data set, make the full set(s) the primary.
14. Re-run Transformations discovery. If you created a new sample set in Step 10 Step 12 the results should be improved from the original results.
15. Return to the Data Sets step and, for the target data set, make the full set(s) the primary.

17. Review Transformations. Improve any expressions with a less-than-satisfactory hit rate.

Apply Sample Set Results to the Full Data Set:

About this task

Once you Perform Discovery on Large Data Sets, apply the discovered transformations to the Full Set sample set (or another sample set that contains all of the data you want to map). Compare the results to confirm that the sample set was representative and produced accurate maps.

To apply transformations to the full data set:

Procedure

1. Process the data up to the point where you want to use the full data set. In most cases, this will be after you have created maps and verified all joins, bindings, transformations, and so on.

2. Display the Data Sets window.

3. Change each file’s sample set from the smaller one used during discovery to the Full Set sample set, or to the sample set containing all of the data you want to map in that table. Update all appropriate files in both the source and the target data sets.

4. Display the Maps Joins window.

5. Click Re-Run Step, then check the Only Refresh Statistics option.

6. Click Run. The statistics in the Joins window are updated based on the primary Join condition, using the data in the full sample sets.

7. Verify that the new Joins statistics are acceptable.
   • If they are, continue with the next step.
   • If not, modify the Joins condition until the statistics are acceptable.

8. Display the Bindings window and refresh the statistics as described above (click Re-Run Step, then check the Only Refresh Statistics option).

9. Check that the binding conditions have not changed. Verify that the expressions and statistics are acceptable, and modify them if necessary.
   Check the refreshed target hit statistics using Show Hits. The results should be as high as or higher than the rates based on the smaller sample set. If not, use Show Misses to examine the misses and determine why the rate has decreased. You may need to discover a filter to improve the statistics.

10. If used, refresh the Where Clause statistics as described above. Verify the results and modify them if necessary.

11. Refresh the Transformations statistics as described above. Verify the expressions and statistics, and modify them if necessary.
   Check the refreshed target hit statistics using Show Hits. If the results are lower than from the smaller sample set, use Show Misses to examine the misses and determine why the rate has decreased. You may need to discover a data rule to improve the statistics.

12. If used, refresh the Reverse Pivots statistics as described above. Verify the expressions and statistics, and modify them if necessary.
Results

The results from the sample sets are now applied to the full data set.

The Sample Set Definition Screen
The Sample Set Definition screen is shown below.

Edit Column Expression:

Click Edit Column Expression to display a popup showing all columns in the selected sample set. The list provides useful metadata about each column, including data type and format, native type, selectivity, and so on.
About this task

For more information, see Changing the Select Statement.

Substitute Column Values (Mask Records):

The Edit Expression field allows you to substitute a string for the values in one or more columns. The string you substitute will replace that column's original values in all discovery, analysis, reporting, and export actions. The original values are not recoverable in any files created from this sample set.

One use for this feature is to specify a valid column expression in the column, such as to_date, to_char, ascii, and so on. You can also specify a SQL expression, or normalize the case with an upper or lower expression.

Another use is to mask sensitive data such as salaries or social security numbers.

You cannot mask records in the Full Set. You can only mask values in sample sets that you create. To apply masked values to the entire table, create a sample set that contains all records in the table, mask any values needed, and mark that sample set as primary.

The original values remain in the sample set. You can restore them at any time by using these instructions to replace the substitute string with the original TABLE_NAME.COLUMN_NAME value for the column.

Changing the Select Statement describes how to substitute column values with a string or function.

Note: If the new value is greater in length than the specified column length, the value may be truncated. Depending on the driver, an error may or may not be logged. For example, the Sybase ODBC driver logs an error when it truncates a value, but the Oracle ODBC driver does not.

Set Columns to Null:

In some projects, it can be useful to set one or more columns to Null if the columns do not contain data relevant to the project. Null columns are not processed during tasks, which results in faster discovery time.

About this task

Note: Only set a column to Null when you are certain it does not contain any data relevant to the project. If you run discovery and later change a column to Null (or revert it from Null to its original values), re-run all discovery tasks.

To set a column to Null, click on the column name in the Column Name field. In the Edit Expression field, type \texttt{null} to replace the existing expression.

If you are changing many columns to Null, it is faster to click Set All Null. This immediately changes the value of all columns in the table to \texttt{null}. You can then select individual columns and change their expressions back to their original values, or any other values. You can also revert all columns back to their original values at any time by clicking Revert All.
Perform Column Analysis

As soon as you start adding database tables to a data set, you can preview each table's contents and column lengths in the Column Analysis window. You can also immediately view the metadata for the native type, data type, and length of database columns.

However, some database tables contain incorrect metadata, and text files do not define the data types. To discover this information, you need to run the Column Analysis step.

When Column Analysis is finished, this window displays the data types of all columns and statistical information for all sources in each data set. At this point, you verify that the data types for all source and target columns are correctly identified, whether specified in metadata or discovered by Discovery. If necessary, you can manually change a column's data type and some other metadata, using data preview to verify the actual data.

Note: Always re-run discovery, including Column Analysis, after importing tables or text files, changing a primary sample set, reloading or reimporting tables, modifying a logical table, or performing any other action that affects the contents of a data set.

Overview

Column analysis is the first step towards data relationship discovery. During column analysis, Discovery performs the following actions:

- **Performing Data Type Discovery**
- **Profiling Data (Column Statistics)**
- "Column Classification" on page 257

These actions are described below. In addition, there may be some Data Types Not Supported or Discovered.

Performing Data Type Discovery

In text files, and often in database sources, numeric and date-time data are often represented as strings (varchar), meaning the defined data type may not accurately indicate the nature of the data. This can affect the discovery output, because the more accurate the data types are, the better the discovered results.

During data type discovery, Discovery scans the data to determine whether a column contains date-time or numeric data. If found, this data is normalized to either NUMBERSTRING or DATETIMESTAMP, allowing related values to be identified and compared.

Discovery also identifies small amounts of defective data. For example, if 99% of the data is numeric and 1% is not numeric because it is contaminated, data type discovery identifies this data and helps you decide whether to tolerate it or correct it. For discovery purposes, it is often acceptable to ignore these values.

**Example 1.** Numeric strings in the raw data are in various formats: -123456, $123456, ($123456.00), 123,456, 123,456.000, 000123456, and so on. During data type discovery, Discovery automatically normalizes these values to the IBM InfoSphere
Discovery number format so they can be compared as part of the discovery process. For example, stripping the leading 0’s is very useful in making the numbers comparable.

Example 2. Date-time strings in raw data are in varying formats: 2/3/82, 3/24/87, 09/4/96, 7/2/1987, 8/11/1988, 07/5/1955, 02/12/1968, and so on. During data type discovery, these formats are normalized to one IBM InfoSphere Discovery date-time format. For example, 2006-07-14 00:00:00.

Profiling Data (Column Statistics)

As part of column analysis, Discovery profiles the data in the source and target columns by calculating a set of statistical properties for each column. The statistics include cardinality, selectivity, non-null value selectivity, null value rates, most common value, whether the column is considered sparse (containing only a few unique values), and value frequencies.

These statistics are computed once, and used by further analysis down the road, such as primary-foreign key analysis, join condition discovery, and source-to-target column matches. These statistics are mostly read-only. Expert users can examine these properties for clues during analysis.

Note: Null and empty values are not included when calculating Cardinality.

Truncating Long Values During Profiling

During profiling, only the first 512 characters in a value are used. This number cannot be changed. Any additional characters are truncated.

Staging Long Values

By default, the first 512 characters in a value are saved in the staging table. You can increase or decrease this number using the global option Maximum number of characters to stage in a column.

Truncating Long Values During Profiling:

During profiling, only the first 512 characters in a value are used. This number cannot be changed. Any additional characters are truncated.

Staging Long Values:

By default, the first 512 characters in a value are saved in the staging table. You can increase or decrease this number using the global option Maximum number of characters to stage in a column.

Identifying Sparse columns:

A sparse column is one that contains mostly the same value except for a few exceptions. Sparse columns happen often and require special treatment during discovery processing.

In the example below, both Col_x and Col_y are sparse. In Col_x, 90% of the values are the same (“null”), and 10% are different (“34”). In Col_y, 80% of the values are the same (“Saratoga”) and the remaining 20% are different (“xy” and “wu”).
Automatic Sampling of High Cardinality Columns

Large tables containing columns with high cardinality can use a large amount of database storage for profiling and other processing tasks. Sampling the data reduces the database space utilization and processing time. However, it can be difficult to accurately sample the data in projects containing large tables with high cardinality columns.

Discovery provides an option that automatically samples the data in high-cardinality columns during the Column Analysis step. The algorithm avoids simple or obvious methods of selecting data, and is based in part on a value you can modify. When you change the value and re-run Column Analysis, the algorithm selects different values. Columns that are not high cardinality are not sampled, and are analyzed in full.

The key feature of automatic sampling is that data is sampled consistently across all high-cardinality columns in all tables, files, and data sets in the project. This preserves the existing cross-column and cross-system relationships. As a result, Discovery can use this greatly reduced sample set but still generally discovers the cross-column relationships, including single-column primary-foreign key relationships.

When column analysis is performed using automatic sampling, the resulting sampled data (from both the sampled and non-sampled columns) is saved and used for most calculations and discovery in the project, such as the Column Matches, Target Matches, and Overlaps tasks. The sampled data set is also used to determine the cardinality, mode, and most other values displayed in grids.

Discovery also creates a second sample set containing all of the values in the data set. This complete sample set is used for other tasks and calculations, such as the hit/miss statistics in the Column Analysis Formats column and the Min/Max values.

A drawback of using a sample set created by automatic sampling (or by most other methods of sampling very large data sets) is that the discovery results may not accurately represent the full data set. The PF Keys screen allows you to view a list of column matches that could have possibly been missed due to automatic sampling. Although it is unlikely that meaningful matches are missed when you use automatic sampling, it is always useful to review the Missed Matches list, particularly if the PF Keys task did not discover the expected number of relationships. If any of the missed matches are meaningful, you can add them as
PF Keys. If no relationships or meaningful matches are found, you may need to modify the High Cardinality or Size Reduction Factor threshold, or re-run discovery with automatic sampling turned off.

It is useful to review the missed matches list to see if any of these matches are meaningful and need to be added to the list or if discovery should be performed on the entire data sets without sampling.

The options that affect automatic sampling are in the Profile section of the option set.

**Column Classification**

Discovery supports the use of column classification. Column classification is an optional step performed during Column Analysis. A classification is a description associated with a column. A column can have more than one classification. Using classifications in your project makes it easier for you to keep track of important or specific columns as you work through the discovery process.

Discovery can attempt to assign classifications to each column during Column Analysis, using algorithms and other methods to evaluate the actual data in each column. If a column meets the criteria for a classification, the classification is automatically assigned to the column during discovery. After Column Analysis, you can change the automatically-assigned classification, if necessary. You can also manually assign classifications to columns and create new classifications.

For example, the column name C5000_2 does not help you identify the type of data in the column. However, if the Column Analysis step assigns the classification of SSN to column C5000_2, the task has determined that C5000_2 contains Social Security numbers. A classification of USPHN would indicate that the column contains United States telephone numbers, VISA would indicate the column contains VISA card numbers, and so on.

Adding a classification to a column does not affect any subsequent Discovery steps. Classification only allows you to easily identify the type of data contained in a column.

Classifications can be automatically discovered during the following tasks:

- “Perform Column Analysis” on page 260
- “Re-run Column Analysis” on page 261
- “Discover PF Keys” on page 288
- “Discover Overlaps” on page 370

Column classification can be manually assigned, changed, and deleted as described in the following sections:

- “Column Classification View” on page 259
- “Assign, Change, or Remove Column Classification” on page 280
- “Manage Columns” on page 277

**Types of Classifications**

Discovery contains a large list of predefined classifications that identify common kinds of data often contained in tables. Examples of predefined classifications are DOB (date of birth), GEN (gender code), PN (personal name), AN (account number), CCN (credit card number), and so on.
Some of the most common predefined classifications have an accompanying set of algorithms and rules that allow the Column Analysis task to determine whether a column meets the criteria for that classification. These predefined algorithmic classifications are the only classifications that can be automatically applied to a column during the Column Analysis task.

However, you can also manually assign classifications, including classifications that are predefined, imported from Business Glossary, or defined manually. Classifications that you define or import from Business Glossary are only available in the current project.

You can define algorithms in Discovery for any imported or defined classification, and also change predefined algorithms.

**Using Classifications**

Classifications provide an easy way to identify the type of data contained in a column without needing to preview the data. If several columns in the project contain the same type of data, they can have the same classification. For example, if the columns HQ.WEST.CFN, EAST.A24, and SHIP.KEY_21 all contain customer first names, they can all be classified as CUST_FIRSTNAME.

You can also use classifications to identify important, sensitive, or questionable columns and track those columns throughout the Discovery process.

A column can have more than one classification. For example, the column C50006.MM can be given the classification I (Identifier) as well as the classification SENSITIVE.

**Automatically Assigned Classifications in Other Steps**

During the Column Matches and Overlaps processing, Discovery checks the classifications assigned in the Column Analysis step, and can automatically classify additional columns based on those classifications. The classification must be marked as Approved in order to be considered for other columns. For example, during Column Analysis, you may have manually defined, assigned, and approved the classification DIV (division) to the column DRME.539. During Overlaps processing, if Discovery finds a high number of overlaps between DRME.539 and a column named D2003.ADR, Discovery will assign the classification DIV to the column D2003.ADR.

**Excluding Columns from Classification**

If you do not want Discovery to automatically assign a particular classification to a specific column, you can mark that classification as excluded for that column. Discovery will not automatically assign that classification to that column during any task execution.

**Changing Classifications**

Column classifications are assigned in the Column Analysis screen and the Column Classification View. The assigned classifications are shown on several of the subsequent Discovery screens, allowing you to easily find columns later in the discovery process.
In the Column Analysis tab, you can change any classification that was automatically discovered and assigned. You can also create new classifications and assign them to appropriate columns.

**Column Classification View**
The Column Classifications window displays the descriptions that are assigned to each column by Discovery, and if they exist, the user-defined classifications.

Selecting **Column Classification View** or clicking **Classification View** in the Column Analysis or Overlaps window displays the **Column Classifications** window. You can display the column classifications in all data sets in the project, or in just one data set. Each project has its own set of column classifications. For more information, see [Column Classification](#).

In addition to the standard tools, the following tools are available in this window:

- **Refresh Statistics** updates the **Hit Rate** shown for the columns of an assigned classification. Expand an assigned classification to view the **Hit Rate**.
- **Show Hits** and **Show Misses** are only available for columns that are assigned to at least one discoverable classification. Expand the column classification to display all columns in the project that are assigned to or excluded from that classification. Click on a column and choose **Show Hits** or **Show Misses** from the drop-down menu. The Preview Criteria window opens. Click **OK** to then display metadata, classification information, and statistics about the column. While in the classification display window, you can again click **Show Hits** or **Show Misses** to change the view.
- **Frequencies** are only available for columns assigned to or excluded from at least one classification. Expand the column and select an entry, then select **Value Frequencies** or **Length Frequency**.
- **Column Classification Report** creates a table containing only the classifications assigned to or excluded from a column. The report can be exported in a variety of formats using the standard **Export Rows** commands.
- **Business Glossary** allows you to import or export Business Glossary terms. For more information, see [Export Business Glossary Terms](#) or [Import Business Glossary Terms](#).
- **Action** is only available for columns assigned to or excluded from at least one classification. The available options depend on whether the column is assigned or excluded. Expand the column and select an entry, then select the appropriate action: **Exclude** the selected column from the classification, **Unclassify** (remove the classification for) the column, **Set Approved** to approve the column classification, or **Reset Approved** to remove the approval.

The **Column Classifications** window contains the following columns:

- **Column Classification** lists the abbreviated name of each currently-defined classification in the project.
- **Full Name** is the full name of each classification.
- **Assigned** indicates how many columns in the project currently are assigned this classification.
- **Excluded** indicates how many columns in the project currently are excluded from this classification.
- **Discoverable** indicates whether the classification can be automatically assigned as a result of a Discovery task, or whether it can only be manually assigned.
Predefined indicates whether the classification was included as a predefined item in Discovery, or was manually defined or imported after the project was created.

Business Glossary Term indicates whether the classification is also used as a Business Glossary term.

Data Types Not Supported or Discovered

When you add a database table to a data set, Discovery checks the data type of each column and imports only the columns containing supported datatypes. Entries in the Output Pane show you the columns that were not imported. For this reason, it is a good idea to keep the Output Pane open whenever you are importing database tables into a data source.

Supported data types are listed in ODBC Driver Data Type Conversion.

Perform Column Analysis

About this task

Running the Column Analysis task performs data type discovery and profiling (statistics) on the source and target data sets. Before performing Column Analysis, you must have at least one table or text file in at least one data set.

Note: You can select to run data type discovery (profiling), explicit staging, or both actions.

To advance to the Column Analysis window without processing the files, click the Column Analysis tab. The Column Analysis window opens displaying the metadata from any database tables that had metadata. No statistics or discovered datatypes are displayed.

Procedure

1. Click Run Next Steps in the Data Sets window. The Processing Options Dialog appears with Column Analysis selected.
2. If desired, drag down the Steps arrow to execute additional steps at the same time.
3. If desired, click Sub Steps and select only certain sub steps for execution. You can customize the sub steps for all selected steps.
   - Data Type Discovery examines the data in each column to determine each column’s data type.
   - Explicit Staging stages or re-stages each table in the staging data source.
   - Column Classification: Algorithms examines the data in each column to determine if a known classification can be applied to the column. See Column Classification.
4. Verify that the task Options are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.
   Note: Modifying task options is an advanced activity. Options are described in Reference of Task Options. Modifying options and creating new option sets is described in Option Sets.
5. Click Run to execute Column Analysis processing. If the procedure does not complete correctly, see Troubleshooting Task Execution.
Use **Re-run Step** in the Column Analysis window if:

- You clicked directly on the Column Analysis tab to display metadata after creating Data Sets, instead of using **Run Next Steps** from the Data Sets window to run the Column Analysis task.
- You ran the Column Analysis task previously, but you manually changed any of the VarChar, DateType or NumberString data types and now want to reprofiles the data (refresh the statistics).
- You changed the primary sample set for one or more logical tables.
- You want to restage the primary sample sets and dependent non-primary sample sets before performing a particular processing task.
Procedure

1. Click **Re-Run Step** in the Column Analysis window. The **Processing Options** Dialog appears. By default, **Data Type Discovery** is selected, and the **All Objects** button is selected.

2. If desired, click **Select** in the Parameters pane and unselect any tables you do not want to be processed in this step.

3. Select the actions to be performed in this step.
   - **Data Type Discovery** runs data type discovery for the selected tables.
   - **Column Classification: Algorithms** is only available when at least one column has an approved classification. See “Column Classification” on page 257.
   - **Stage Selected Items** displays the logical tables, sample sets, and physical tables in the project, with each logical table’s primary sample set selected by default. If you click Select to customize the list, the primary sample sets are green and the physical tables are red. Non-primary sample sets that are unchecked will be restaged if they are dependent on a checked sample set.
   - **Only Refresh Statistics** refreshes statistics for the selected tables without performing profiling.

   **Note:** You cannot run **Only Refresh Statistics** when any other actions are selected. Run the other actions first, if needed, and then run **Only Refresh Statistics** as a separate step.

   a. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   **Note:** Modifying task options is an advanced activity. Options are described in [Reference of Task Options](#). Modifying options and creating new option sets is described in [Option Sets](#).

   b. Click **Run**.

Results

Although the next window is displayed immediately, it may take some time for the actual data processing to complete. Check the **Activity Viewer** for the status of the task.

Review Column Analysis Data

About this task

In the Column Analysis window, verify that the data shown is correct and valid for each user data file in both data sets.

Procedure

1. Click the appropriate data set tab to display the source or target data sources in the **Column Analysis Tables Pane**.

2. For each data set, click each table in the **Tables** pane to populate the grid with that table’s information. Verify that the information is correct and valid for each file, and make changes as required.

   For example, if most of the data in a column of employee IDs is in the format “07aabb”, where all aa values are between 01-12 and all bb values are between...
01-31, the column may be incorrectly discovered as DATETIMESTAMP. You would change the column format to NUMBERSTRING or VARCHAR as appropriate.

3. If using classification, verify the classification of each column and "Assign, Change, or Remove Column Classification" on page 280 as appropriate.

4. Check the Value Frequencies of columns as needed.

5. As needed, use Column Analysis Data Preview to display the source data of the selected user data file.

6. When you have approved a column, check the Approved box. This locks all settings for that column, and keeps the data in the column from changing if you re-run Column Analysis or refresh the statistics. (You can always go back and uncheck this if you want to make further changes.)

**Change Data Type Formats**

**About this task**

You can modify the format of NumberString, Date, DateTime, and DateTimeString data types in the Format column of the Column Analysis Window.

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Precision</th>
<th>Scale</th>
<th>Formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>7</td>
<td>38</td>
<td>0</td>
<td>-NNNN[38],[NNNN[38]]</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>String</td>
<td>6</td>
<td>38</td>
<td>0</td>
<td>-NNNN[38],[NNNN[38]]</td>
</tr>
<tr>
<td>String</td>
<td>4</td>
<td>38</td>
<td>0</td>
<td>-NNNN[38],[NNNN[38]],-N,NNN[38],[N,NNN[38]]</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
</tbody>
</table>

The example Formats dialog above shows two formats for NumberString data types in the EMPLOYEE_ID column.

To view details about a format, click in a cell listing a formats and then click the icon to display the Formats dialog. This dialog contains an entry for each format discovered in the selected column.

In addition to the standard icons and functions, the fields and links on the Formats dialog are:

- **Format**
- **Hits** updates the Hits statistic for that format.
- **Positive Sample**
- **Negative Sample**
- **Refresh Statistics** after adding, deleting, or modifying a format, click this icon to update all of the Hits values, Show Hits, and Show Misses.
• **Unmatched Rows**
  
  • **+** Adds a new blank line. Type the new format in the line or build the format from the dropdown menu in the line. See [Change the NumberString Format](#) or [Change the Date, DateTime, or DateTimeString Format](#).
  
  • **-** Deletes the selected format from the grid.
  
  • **↑** Moves the selected format up one row in the list.
  
  • **↓** Moves the selected format down one row in the list.

**Hits**

Clicking a value in the **Hits** column displays a popup showing all values in the column that can be read using that format. All rows may be readable using one format, or several formats may be necessary to achieve a total of 100% for the column.

**About this task**

**Show Rows**

Clicking **Show Rows** displays a popup showing all rows in the column that use any of the listed formats. For example, if two formats are listed, and half of the rows use one format and the other half uses the other format, this popup contains all of the rows in the table.

**About this task**

If no values are shown in this popup, none of the rows in the column match any of the listed formats.
Unmatched Rows
About this task

Clicking Unmatched Rows displays a popup showing all rows in the column that do not use any of the listed formats.

If no values are shown in this popup, all of the rows in the table match at least one of the listed formats.

Change the NumberString Format
About this task

To edit a NumberString format:
Procedure
1. Display the Formats dialog by left-clicking in the appropriate column's Formats cell, then clicking the \( \ldots \) icon.

2. In the Formats popup, click on an entry in the Format column. The Edit Number String Format dialog appears.
The Edit Format dialog displays the properties of the selected NumberString data type. If the selected data type does not use a particular property, the corresponding field is blank.

3. Modify the properties as required. Each property is described below.
   - **Precision**: Overall level of precision of the data, expressed as the total number of places on both sides of the decimal point.
   - **Scale**: Level of Decimal Precision of the data, expressed as the number of places to the right of the decimal point.
   - **Decimal Separator**: Character used to indicate the decimal point; ordinarily a period, but you can use any other character.
   - **Thousand Separator**: Character used to separate the hundreds from the thousands place; ordinarily a comma, but you can use any other character.
   - **Start String**: Any string of characters you want to appear at the beginning of every value, such as a currency symbol. Select the dollar, euro or yen sign, or type any other string.
   - **End String**: Any string of characters you want to appear at the end of every value. Open field; supply any string.
   - **Negative Sign**: Specify the mode of marking values as negative: only valid choices are minus sign or parentheses.

4. When you are finished, click **OK** to close the Edit Format dialog. Edit any other formats as required.
5. When you have finished editing all formats, click **Done** in the Formats popup.

### Change the Date, DateTime, or DateTimeString Format

**About this task**

This section describes changing the format of individual columns after importing tables and running Column Analysis and datatype discovery. If the default Discovery formats for date, datetime, and datetimestring are not appropriate or sufficient for your data, you can modify the ScanFormats.xml file before importing data into Discovery.

Each column identified as a Date, DateTime, or DateTimeString data type in the **Format** column was identified during Column Analysis discovery as containing one or more Microsoft .Net DateTimeString formats.

When Discovery stages the data sources, it converts the DateTimeString values from the format(s) listed to the internal Discovery datetime format, **yyyy-MM-dd HH:mm:ss**.

**Note:** The internal Discovery datetime format, **yyyy-MM-dd HH:mm:ss**, cannot be changed and is not affected by the data types or formats shown in the Column Analysis Window.

The **Formats** column below shows that Discovery discovered the values in the **DATE_OF_BIRTH** column use the format **yyyy-MM-dd HH:mm:ss**. Examples of actual data from the table are shown in the **Min** and **Max** columns.

<table>
<thead>
<tr>
<th>Data</th>
<th>Column Name</th>
<th>Date Type</th>
<th>Formats</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>BEGIN_DATE</td>
<td>TIMESTAMP</td>
<td>17</td>
<td>1999-01-10 00:00:00</td>
<td>2002-11-11 00:00:00</td>
</tr>
<tr>
<td></td>
<td>CURRENT_STATUS</td>
<td>VARCHAR</td>
<td>2C</td>
<td>1998-12-28 00:00:00</td>
<td>1995-08-13 00:00:00</td>
</tr>
<tr>
<td></td>
<td>DATE_OF_BIRTH</td>
<td>TIMESTAMP</td>
<td>21</td>
<td>1998-12-28 00:00:00</td>
<td>1995-08-13 00:00:00</td>
</tr>
<tr>
<td></td>
<td>EMPLOYEE_ID</td>
<td>VARCHAR</td>
<td>15</td>
<td>1985-01-01 00:00:00</td>
<td>2005-12-31 00:00:00</td>
</tr>
<tr>
<td></td>
<td>EMPLOYEE_LAST</td>
<td>VARCHAR</td>
<td>50</td>
<td>1985-01-01 00:00:00</td>
<td>2005-12-31 00:00:00</td>
</tr>
<tr>
<td></td>
<td>END_DATE</td>
<td>TIMESTAMP</td>
<td>7</td>
<td>2000-12-31 00:00:00</td>
<td>2004-12-31 00:00:00</td>
</tr>
<tr>
<td></td>
<td>SALUTATION</td>
<td>VARCHAR</td>
<td>6</td>
<td>1985-01-01 00:00:00</td>
<td>2005-12-31 00:00:00</td>
</tr>
<tr>
<td></td>
<td>SSN</td>
<td>VARCHAR</td>
<td>14</td>
<td>1985-01-01 00:00:00</td>
<td>2005-12-31 00:00:00</td>
</tr>
</tbody>
</table>

The DateTimeString is made up of two components: the datestring and the timestring. Both are case-sensitive.

- The **datestring** defines the way the month, date and year are represented in date strings.
- The **timestring** defines the way the hours, minutes and seconds are represented in time strings.

Your data may use more formats than shown here.

If, for example, you ran discovery on a sample set that only included datetime values in one format, but the full table includes datetime values in an additional format, Discovery would not discover that additional format. In that case, you need to manually add the second format here.
Or, if you want to exclude all datetime values of a certain format, you can delete that format here and Discovery will not read values in that format in future discovery.

To change the format of a Date, DateTime, or DateTimeString data type:

**Procedure**

1. Display the Formats popup by left-clicking in the appropriate column's Formats cell, then clicking the icon.

    ![Formats for DATE_OF_BIRTH](image)

2. In the Formats popup, click on an entry in the Format column. The Edit Date/Time Format dialog appears.

    ![Edit Date/Time Format](image)

3. Delete the existing values in the Date/Time string field.
4. Click the right arrow and select Date Format to display a list of Microsoft.Net datestring formats.

5. Click the appropriate datestring format to select it. The format is inserted into the Date/Time string field.

6. Click the right arrow again and select the Time Format list, then select the appropriate time format. You do not need to enter a space between the date format and the time format; Discovery adds one automatically.

7. When the date and time formats are correct, click OK to close the Edit Date/Time Format dialog.

8. In the Formats popup, click Done.

**Change Other Formats**

Values in several other columns, including Classification and Scale, can be configured or modified from the discovered values. After selecting or entering a new value, press Enter or tab out of the cell to commit the change.

**The Column Analysis Window**

The Column Analysis window displays both imported and discovered information about the tables, including metadata, discovered data types, and other statistics about the source and target data sets. Tables are grouped by data set.

[Common Window Elements](#) describes the menus, tabs, and buttons (like Validate Step and Activity Viewer) that are common to most windows.
Column Analysis Tables Pane
The Column Analysis Tables pane displays each database table and text file in the source and target data sets. Click on an entry in this pane to display the corresponding data in the main grid.

Column Analysis Data Grid
The header above the Column Analysis data grid displays the name, primary sample set, and number of rows in the selected table.

Column Analysis Data Grid Tools
The following tools are available in the Column Analysis Data Grid.

- Value Frequencies
- Pattern Frequencies
- Length Frequencies
- Column Classification
- “Manage Columns” on page 277

Metadata:
- Approved
- CDE (available from Column Chooser in TD projects)
- Classification (Group Names) (available from Column Chooser in TD projects)
- 
- Column Name
- Native Type
- Data Type
- Length
- Precision
- Scale
- Format

Statistics:
- Cardinality
- Min
- Max
- Mode
- Mode%
- Sparse
- Null Count
- Notes
  - Blank Count (available from Column Chooser)
  - Selectivity (available from Column Chooser)
  - Non-null Count (available from Column Chooser)
  - Non-null Selectivity (available from Column Chooser)

Column Analysis Data Grid Tools:

The following standard tools are available for viewing and manipulating the data in the Column Analysis Data Grid. Also see the Grid Panes description for common actions you can perform in a grid.
- Advanced Search
- Search
- Auto Filter Row
- Column Chooser
- Export Grid Content
- Value Frequencies
- Pattern Frequencies
- Length Frequencies

The first columns in the grid allow you to designate columns in the table as Critical Data Elements (CDEs) and enter Classification (Group Names). The next group of columns displays Metadata about the selected table, and the remaining group of columns displays Statistics about the selected table.

Several additional columns are available for view, but are not initially shown in the table. For more information, see Column Chooser.
You can modify some of the results in the grid, regardless of whether the information was imported from the data or was discovered by Discovery.

**Note:** Any unapproved changes you make will be overwritten if you re-run Column Analysis. To avoid this, approve the changed rows you wish to keep using the **Approved** checkbox.

Normally, you only need to change metadata that you know has been incorrectly discovered, like a NumberString type that should be a DateTimeString type. However, you may also modify values that are correct but need to be changed for another reason, such as increasing a column’s length.

### Value Frequencies

*About this task*

To display value frequencies, click in a column in the **Column Analysis Data Grid** to select it, and then click **Value Frequencies**. This popup displays a summary of the value frequencies in the selected column. The Data Preview pane at the bottom of the Value Frequencies popup appears after you click **Preview Data**.

In the example below, the column CRM_BRCH_1A.GENDER contains two values. The value \( F \) appears 57 times in the column, while the value \( M \) appears 29 times. The \( F \) row is selected in the Frequencies Pane, so all rows with this value are displayed in the Data Preview pane.
You can also **Select Preview Criteria**.

The following standard tools are available for viewing and manipulating the data in the Value Frequencies popup. Also see the **Grid Panes** description for common actions you can perform in a grid.

- Advanced Search
- Search
- Auto Filter Row
- Column Chooser
- Export Rows

**Pattern Frequencies**

**About this task**

The Pattern Frequencies tool displays a summary of the pattern frequencies in the selected column. To display pattern frequencies, click in a column in the **Column Analysis Data Grid** to select it, and then click **Pattern Frequencies**. This popup lists each value pattern in the selected column and states how often each pattern appears. The Data Preview pane at the bottom of the Pattern Frequencies popup appears after you click **Preview Data**.
In the example below, the column CRM_BRCH_1A.MIDDLE_NAME contains seven patterns. The shortest pattern, AAAA, appears seven times in the column, while the longest pattern, AAAAAAAAAA, appears once. The Data Preview pane below displays all columns with the selected pattern, AAAA.

You can also Select Preview Criteria.

The following standard tools are available for viewing and manipulating the data in the Pattern Frequencies popup. Also see the Grid Panes description for common actions you can perform in a grid.

- Advanced Search
- Search
- Auto Filter Row
- Column Chooser
- Export Rows
Length Frequencies

About this task

The Length Frequencies tool displays a summary of the value lengths in the selected column. To display length frequencies, click in a column in the Column Analysis Data Grid to select it, and then click **Length Frequencies**. This popup lists each value length in the selected column and states how often each length appears. The Data Preview pane at the bottom of the Length Frequencies popup appears after you click **Preview Data**.

In the example below, the column CRM_BRCH_1A.MIDDLE_NAME contains values with seven different lengths. The shortest value length, 4, appears seven times in the column, while the longest value length, 10, appears once. The Data Preview pane displays all columns with the selected length, 9.

You can also **Select Preview Criteria**.

The following standard tools are available for viewing and manipulating the data in the Length Frequencies popup. Also see the **Grid Panes** description for common actions you can perform in a grid.

- **Advanced Search**
Manage Columns displays certain information about all columns in the current data set or all data sets. It also allows you to modify some characteristics of the columns.

About this task

The column classifications can be modified at any time using this window. Instructions are below.

In the Master field, you can display either the current data set, or All Data Sets to show all columns in all data sets. You can also select the Column Groups view or Logical Tables view.

The following information is displayed in the Logical Tables view. Most of the fields are read-only. Some columns are not displayed by default and are available in the Column Chooser:

- Data set Name: Name of the data set containing the logical table.
- Logical Table Name
- Modified: Date the logical table was last modified.
- Remove Duplicates: Indicates whether the Remove Duplicates option is checked in the Logical Table Definition screen.
- Profile Row Count: Number of rows in the logical table.
- Columns: Number of columns in the logical table.
- Foreign Keys: Number of foreign keys in the logical table.
- Sample Set Count: Number of sample sets created for the logical table.
• Primary Sample Set: Name of the primary sample set.
• Note: Allows you to enter a note about this logical table.
• SQL Clauses, Table or Join: Physical table
  • Where
  • Group By
  • Having

The following read-only information is displayed in the Column Groups view.
• Column Name
• Native Type
• Data Type
• Tables with Column: Number of logical tables containing columns with this exact name.
• Assigned: Classifications, if any, assigned to this column name.
• Excluded: Classifications, if any, from which this column name is excluded.

The Details pane shows the following information about the item selected in the upper pane, for both views. Some columns are not displayed by default and are available in the Column Chooser.
• Data Set Name
• Logical Table Name
• Column Name
• Table Column Name: Full column name, in TABLE.COLUMN format.
• Has SS: Indicates whether the column is used in a sample set.
• Data Type
• Overlap Columns: Full name of other columns containing overlapping values with this column.
• Length
• Precision
• Scale
• Assigned
• Excluded
• # Approved: Number of classifications that are approved for this column
• Where Clause Is This:
  • Logical Table
  • Primary Sample Set
  • Sample Set Name
  • Cardinality
  • Selectivity
  • Min
  • Max
  • Range
  • Mode
  • Null Count
  • Non Null Rows
  • Non Null Selectivity
Procedure
1. In the Details pane, select one or more columns.
2. Click Modify Columns. The Column Properties dialog appears.
3. In the Assign Classifications row, click the dropdown arrow in the Value column and select a classification to apply to the selected columns, or select Clear Classification to remove all classifications currently assigned to the selected columns. You can also exclude the selected columns from a classification, or clear all current excluded classifications.
4. Click OK to accept the changes and close the dialog.

Exporting a table to the staging database
You can export a table and columns from the InfoSphere Discovery Column Analysis view along with its metadata to a staging database that is defined with your project.

Procedure
1. Click the Column Analysis tab and then select a data set and a table that you want to export. This export process assumes that you have already designated certain columns as critical data elements (CDEs)
2. Select the Manage Columns drop-down menu and click Export Columns to Staging.
3. In the Export to Staging window, use the drop-down list to specify an existing object name in the ObjectName field to identify this table or group of tables that you might later append. To create an object name, type over the current value with any alphanumeric characters in the Value field.
4. In the DbTableName field, use the drop-down list to specify an existing name for this exported table. To create a table name, type over the current value with any alphanumeric characters in the Value field. If the table name exists, you can decide to append to the existing table or to drop the current table and recreate the table by selecting the appropriate value in the DropIfExists field.
5. Click the Value field next to **DropIfExists** and select the appropriate value depending on how you want InfoSphere Discovery to treat an existing table.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>If the table exists, append the values to the existing table.</td>
</tr>
<tr>
<td>true</td>
<td>If the table exists, drop the current table, and then recreate the table.</td>
</tr>
</tbody>
</table>

A timestamp in the output indicates the current time of the export to help identify the variations in the appended tables.

6. Click **OK** to complete the export to the staging database. You can export one table at a time.

**Assign, Change, or Remove Column Classification**

This procedure describes how to manually assign a classification to a column. It also describes how to manually exclude columns from classification and how to change an automatically-assigned classification.

**About this task**

You can manually define and assign column classifications before or after executing the Column Analysis task. If you selected Classification when executing the Column Analysis task, review those classifications now, and use this procedure to change any classifications that may be incorrect.

If you do not want Discovery to automatically assign a particular classification to a specific column, you can mark that classification as excluded for the column. Discovery will not automatically assign that classification to that column during any task execution.

For example, your data may have a column named CUST.ID7 containing verified work email addresses, and a column named CUST.EM2 containing unverified email addresses. You define a new classification named WORK_EMAIL and assign it to CUST.ID7, but do not classify CUST.EM2 because it is not important in the current project. If Discovery finds many overlapping values between CUST.ID7 and CUST.EM2 during the Overlaps task, it may automatically assign the classification WORK_EMAIL to CUST.EM2. You can prevent this from happening by marking WORK_EMAIL as an excluded classification for the CUST.EM2 column.

All user-defined classifications are considered for possible assignment during the Column Matches and Overlaps tasks. However, Discovery only considers other classifications (predefined, algorithmic, and Business Glossary) for possible assignment if those classifications have already been assigned to a column and marked as approved.

For example, during Column Analysis, Discovery may have automatically assigned the algorithmic classification EA (email address) to the column CUST.ID7, but not assigned a classification to CUST.EM2 because not enough of the records in CUST.EM2 were in the correct email format. If you mark the EA classification for CUST.ID7 as approved, and Discovery determines during the Overlaps step that there is a high level of overlap between CUST.ID7 and CUST.EM2, Discovery will assign the EA classification to CUST.EM2. However, if you do not mark the EA classification for CUST.ID7 as approved, Discovery will not assign EA to CUST.EM2 during Overlaps.
Procedure

1. In the main data grid of the Column Analysis tab, click in the Classification cell for the desired row. The cell may be empty or it may already contain a classification.

2. Click the ellipses (...) button that appears in the cell. The Classifications for `<column_name>` screen opens and displays the list of classifications in this project. The Status column is empty for unclassified columns.

3. To assign or change one classification for this column, click the Status cell in the appropriate classification row and select the appropriate status from the dropdown menu:
   - **Assigned** assigns this classification to the column. The type of classification (Discovered, User Specified, Propagated via Matches and Propagated via Overlaps) appears in the Method column. If this is a Predefined classification with an algorithm, the number of values in the column that meet the criteria are shown in the **Hits** column.
   - **Excluded** prevents Discovery from automatically assigning this classification to the column in any task.
   - `<Clear>` removes an assigned or excluded classification from the column. Discovery can automatically assign this classification again if it determines the classification is appropriate.

4. Alternatively, you can assign or change several classifications at the same time by shift-clicking or control-clicking the appropriate classifications and then clicking **Assign**, **Exclude**, or **Clear Selected** in the menu bar.

5. Click **Approved**, if appropriate, for classifications. An approved predefined classification will not be removed during any Discovery processing. An approved user specified or imported classification will also not be removed, and may be assigned in the Column Matches and Overlaps tasks to other columns that have a high overlap with this column.

6. Click **Assign** or **Exclude** to apply the selected classification changes to the column.

7. When all changes are made, click **OK** to close the Classifications for `<column_name>` screen.

Importing Business Glossary XML

You can import Business Glossary XML.

Procedure

1. From the menu bar click **Project > Import > Business Glossary XML**.

2. In the browse window, find the specific IBM Business Glossary XML exported file that you want to import, and click **Open**.

3. You can now select the terms and assets to import in the **Import Business Terms and Database Column Assignments** window.

Import IBM Business Glossary Terms and Assigned Assets

You can import business terms and assigned assets from IBM Business Glossary into the current project. The imported terms are automatically assigned to existing columns that exactly match the imported assigned assets. They can also be manually assigned to other columns in the project in the Column Analysis tab. Discovery Studio classifications can also be exported into Business Glossary.
About this task

Each classification name in the Discovery project must be unique. If any term in the glossary exactly matches an existing Discovery classification in the current project, the term is not imported.

After a glossary has been imported, any new assignments made in Business Glossary will be ignored when the glossary is re-imported into Discovery Studio. In order to use the new assignments, delete the corresponding Discovery Studio classification before re-importing the glossary. Note that when you delete a classification in Discovery Studio, it is removed from all assigned columns and must be re-applied to each appropriate column.

It is a good practice to consider Business Glossary as the main repository of business terms. When new classifications and assignments are made in Discovery Studio, you should export the new term-to-column assignments to Business Glossary, delete the classifications in Discovery Studio, then re-import the glossary into Discovery Studio.

Procedure
1. In the Business Glossary web console, export a full or partial glossary as an XML file, making sure the Assigned Assets option is checked. Copy the file to a directory that can be access by Discovery Studio.
2. In Discovery Studio, open a project and select View>Classification View to display the Column Classification window.
3. Click Business Glossary and select Import Terms & Assignments.
4. Click Browse and navigate to the appropriate directory and select the XML file, then click OK.
5. In the Import Business Terms window, select the terms and assets to import. Terms that already exist as Discovery classifications in this project are greyed out, and will not be imported.
6. Click Import. The terms are added to the list of classifications.
7. If necessary, delete any imported terms. You can now use the imported terms as column classifications.

Export IBM Business Glossary Terms and Assigned Assets
Discovery allows you to export column classifications as IBM Business Glossary terms and assigned assets.

About this task

It is a good practice to consider Business Glossary as the main repository of business terms. When new classifications and assignments are made in Discovery Studio, you should export the new term-to-column assignments to Business Glossary, delete the classifications in Discovery Studio, then re-import the glossary into Discovery Studio.

Procedure
1. In Discovery Studio, open a project and select View>Classification View to display the Column Classification window.
2. Click Business Glossary and select Export Assignments. The Export Business Terms and Assignments window appears.
3. Expand the list to show all terms and assignments. All items are selected by default.
4. Uncheck (deselect) the terms or assignments that will not be exported.
5. Click Browse to navigate to an appropriate directory and select or enter a name for the output file.
6. Click Export. The Export Business Terms and Assignments window closes.
The export file can now be imported into Business Glossary.

**Critical Data Elements (CDEs)**

Critical Data Elements, or CDEs, are an optional designation for individual columns that identify them as important in the project. You can use Classification (Group Names) in conjunction with CDEs to further organize and prioritize columns. Several screens allow you to filter the grid based on CDE designation and classification.

Marking a column as a CDE does not affect any processing or results; it simply allows you to quickly identify the column in subsequent screens.

There is no restriction on the number or types of columns you can mark as CDEs. When you mark or unmark a column as a CDE, the change is immediately reflected in all screens displaying CDEs, such as The Overlaps Window.

The following screenshot shows several rows that have been given CDEs, classifications, or both.
**Column Analysis Data Preview**

**About this task**

The Column Analysis Data Preview pane displays the contents of the selected table.

To display the data, select a table in the Column Analysis Tables Pane and then click **Preview Data**. You can also **Select Preview Criteria**.

---

**Discover PF Keys**

In many relational databases, the primary-foreign key relationships are defined in the metadata and are imported into Discovery when you import the tables into the data set. However, some databases do not define primary-foreign key relationships, and text files cannot include primary-foreign key metadata.

In this step, Discovery discovers column matches. A column match is a relationship between the data in two columns in different tables in the same data set. The relationship may be strong or weak. Discovery has minimum hit rate statistics and other criteria that a column pair must reach in order to be considered a column match.

Based on additional discovery, Discovery promotes certain column matches to the status of primary-foreign keys, or PF Keys. This is a major step in the mapping process because Discovery relies heavily on PF Keys to identify table relationships during the later analysis and mapping steps.

**Overview**

Primary-foreign key relationships are a way for data modelers to encode critical relationships between database tables, and for database management systems to enforce these relationships and to control the integrity of the data. Primary-foreign keys are also the basis for join conditions.

When normalizing a table or modeling many-to-many relationships between entities, data analysts identify primary and foreign keys. These relationships carry semantics that are critical for mapping purposes. If these relationships are declared as part of the database table definition, and if the source or target JDBC driver supports this feature, Discovery will import the primary-foreign key definitions and use them in subsequent discovery.

However, declaring primary-foreign keys within a database schema can carry a performance penalty, so these constraints are often enforced outside of the DBMS. As a result, many of the source and target tables imported into Discovery do not...
explicitly declare primary and foreign keys. Also, text data sources do not contain a schema, so primary-foreign key information cannot be included.

Primary and foreign keys can also be incomplete or partial keys for various reasons, including data contamination. For example, a contaminated primary key may have 10 duplicate values out of 10,000, and a contaminated foreign key may have several odd values that violate the referential constraint.

**Primary-Foreign Keys and PF Keys**

Discovery uses a form of inclusion dependency in the following form:

\[
\text{Child.PI} \ni \text{Parent.ID}
\]

The right hand side of this inclusion may be a natural key, a generated key, or an explicitly declared primary key. If each value in the right hand side is unique (a selectivity of 1), the inclusion is a potential foreign key. While the relationship may be incidental, it looks exactly like a primary-foreign key relationship.

Since the most well-understood example of such inclusion dependency is the primary-foreign key relationship, this is referred to as a **PF key** in Discovery. The right hand side is the **primary** side, and the left hand side is the **foreign** side. The following table compares Discovery's concept of PF keys to the traditional concepts of primary keys and foreign keys.

<table>
<thead>
<tr>
<th></th>
<th>Uniqueness of Primary Key</th>
<th>Number of Primary Keys per Relationship</th>
<th>Foreign Side Conformance</th>
<th>Composite Keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional Primary-foreign Key</td>
<td>Guaranteed unique by DBMS</td>
<td>Maximum of 1</td>
<td>Guaranteed by DBMS to be 100%</td>
<td>Supported</td>
</tr>
<tr>
<td>Discovery PF Key</td>
<td>Primary side may be partial key, allowing Discovery to discover keys from contaminated data</td>
<td>More than 1 because Discovery also considers natural keys and column matches</td>
<td>May be partial</td>
<td>Not discovered by default, but can set option to discover</td>
</tr>
</tbody>
</table>

**Composite PF Keys**

Composite PF keys, also called *multi-column primary-foreign keys* or *composite keys*, are primary-foreign key pairs consisting of more than one column in the primary table and more than one column in the foreign table. The primary and foreign keys contain the same number of columns; for example, a composite PF key can contain two primary columns and two foreign columns, or three primary columns and three foreign columns, but it cannot contain two primary columns and three foreign columns.

Discovering composite PF keys requires more time and system resources than discovering single-column PF keys. For this reason, only single-columns PF keys are discovered by default. If the default discovery does not find an acceptable PF key, set the **Discover Composite Primary-Foreign Keys** option to True. To save time, use the default results to identify tables which are likely to contain composite PF keys, and select only those tables for processing during the task.

**Note:** Discovery does not search for composite PF keys using the same tables and in the same direction as existing tables with *Approved PF Keys*. For example, if
T1.C1=T2.C2 is an approved PF Key where T1.C1 is the primary column and T2.C2 is the foreign column, Discovery will not search for a composite PF Key using T1 as the primary side and T2 as the foreign side. However, it will search for a composite key where T2 is the primary side and T1 is the foreign side.

**How Discovery Uses PF Keys**

Discovery uses PF keys in two main ways:

- Most join conditions used in source-to-target mapping are based on PF keys. Identifying PF keys allows Discovery to automatically discover correct join conditions, expanding the scope of discovered maps. For example, assume source tables S1 and S2 with a PF key relationship, and a target table, T. Discovery can discover a map of S1, S2 ‡ T.

- PF keys allow Discovery to intelligently make use of known lookup tables (*data rules*) for mapping analysis. For example, assume the following is known:
  - S.Mark translates to T.Score with a lookup table.
  - Two other columns (S.X and S.Y) reference S.Mark.
  - T.Y references T.Score.

Discovery knows that S.X and T.Y might use the same lookup table for mapping. If Discovery determines that S.X and T.Y are semantically the same, it will recognize that a known lookup table exists that may be helpful for source-to-target transformation.

**PF Key Statistics**

Discovery computes the following statistics for each column in a PF Key.

- **Selectivity** (the **Cardinality** of the column divided by its row count).
  
  A high Selectivity means that most of the values in the column are unique. Low selectivity means most values are duplicates. Valid PF Keys usually have high Selectivity on each side. When both keys have low Selectivity, the match is probably an incidental value overlap.

- **Row Hit Rate** the number of rows in column A that have corresponding values (hits) in column B, divided by the total number of rows in column A.
  
  A high Row Hit Rate (above 90%) indicates a one-to-one relationship where most of the values in the key are also in the other key. A low Row Hit Rate indicates most of the values in this key don't appear in the other key. In a valid PF Key, one or both sides usually have a high Row Hit Rate. However, a valid primary key can have a low Row Hit Rate if it also has high Selectivity. This combination means the primary key has many unique values and the foreign key contains only a few of those values.

- **Value Hit Rate** the number of unique values in column A with corresponding values (hits) in column B, divided by the **Cardinality** of column A.
  
  A high Value Hit Rate (above 90%) indicates most of the unique values in the key are also in the other key. A low Value Hit Rate means most of the unique values don't appear in the other key.

  In a valid PF Key, one or both sides usually have a high Value Hit Rate. A valid PF Key can have a low primary key Value Hit Rate and a high foreign key Value Hit Rate; this indicates the primary key contains most of the foreign key's unique values along with many more unique values.

These statistics help you focus on relationships that are potentially meaningful, and to assess whether a relationship is useful or valid. A 'perfect' PF Key has a primary key Selectivity of 1 with the foreign key having a Row Hit Rate and a Value Hit
Rate of 100. However, remember to always check the Data Preview before determining whether a PF Key is valid. This displays the actual rows that match or do not match the expression.

**PF Key Examples**

The following examples are helpful in understanding PF Key results.

**PF Key Example 1:**

In a set of two tables, Discovery has identified the PF Key
COUNTRY.ID=REGION.CTRY. COUNTRY.ID is the primary key.

<table>
<thead>
<tr>
<th>Table 10. Frequency</th>
<th>Row Hit Rate - CT_ID</th>
<th>Row Hit Rate - RG</th>
<th>Value Hit Rate - CT_ID</th>
<th>Value Hit Rate - RG</th>
<th>Selectivity - CT_ID</th>
<th>Selectivity - RG</th>
</tr>
</thead>
<tbody>
<tr>
<td>RG.CTRY -&gt; CT_ID.ID</td>
<td>83% (5/6)</td>
<td>100% (9/9)</td>
<td>83% (5/6)</td>
<td>100% (5/5)</td>
<td>100% (6/6)</td>
<td>56% (5/9)</td>
</tr>
</tbody>
</table>

Looking at the Row Hit Rate, the primary key, CT_ID.ID, has 6 rows with 5 of the values having hits in RG.CTRY. RG.CTRY has 9 rows, and all of those values have hits in CT_ID.ID. This means most of the values in CT_ID.ID are in RG.CTRY, which indicates the PF Key may be valid. RG.CTRY contains some duplicate values, which is acceptable in a foreign key.

The Value Hit Rate reveals that all of the unique values in CT_ID.ID and RG.CTRY appear in the other column also.

The Selectivity reveals that each row in CT_ID.ID has a unique value, but about half of the values in RG.CTRY are duplicates. This is a common pattern when the foreign key is a subset of the primary key. Based on the statistics, this may be a valid PF Key.

Looking at the Show Hits display, we see that the data supports the conclusion that the PF Key is valid.

<table>
<thead>
<tr>
<th>Table 11. Values in CT_ID and RG</th>
<th>CT_ID Name</th>
<th>CT_ID ID</th>
<th>RG Region</th>
<th>RG Ctry</th>
<th>RG ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canada</td>
<td>A</td>
<td>Utah</td>
<td>C</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>India</td>
<td>B</td>
<td>Durango</td>
<td>D</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td>C</td>
<td>Hunan</td>
<td>F</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>Mexico</td>
<td>D</td>
<td>Punjab</td>
<td>B</td>
<td>312</td>
<td></td>
</tr>
<tr>
<td>Taiwan</td>
<td>E</td>
<td>Delhi</td>
<td>B</td>
<td>258</td>
<td></td>
</tr>
<tr>
<td>China</td>
<td>F</td>
<td>Texas</td>
<td>C</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ottawa</td>
<td>A</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>New York</td>
<td>C</td>
<td>213</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Taipei</td>
<td>F</td>
<td>120</td>
<td></td>
</tr>
</tbody>
</table>

**PF Key Example 2:**

Let’s look at another example.
In another two tables, Discovery has identified the PF Keys RG.ID=SHIP_REF.WT, with RG_ID as the primary key.

Table 12. Frequency

<table>
<thead>
<tr>
<th>RG -&gt; SHIP_REF - Expression</th>
<th>Row Hit Rate - RG</th>
<th>Row Hit Rate - SHIP_REF</th>
<th>Value Hit Rate - RG</th>
<th>Value Hit Rate - SHIP_REF</th>
<th>Selectivity - RG</th>
<th>Selectivity - SHIP_REF</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHIP_REF.WT=RG.ID</td>
<td>100% (5/9)</td>
<td>56% (5/9)</td>
<td>100% (5/5)</td>
<td>100% (9/9)</td>
<td>56% (5/9)</td>
<td></td>
</tr>
</tbody>
</table>

The Row Hit Rate indicates that the primary key, RG.ID, has 9 rows with 5 of them having hits in the foreign key. The foreign key also has 9 rows, with all of the values having hits in the primary key. The low primary Row Hit Rate can still indicate a valid PF Key, if it has a high selectivity, and in this case we see the selectivity for the primary key is 100%.

The Value hit Rate shows that the primary key has 9 unique values, and 5 of them are also in the foreign key. The foreign key has 5 unique values, and all of those are in the primary key. This is also a good indicator that this is a valid PF Key.

Let's look at the data to make sure.

Table 13. Data on RG and SHIP_REF

<table>
<thead>
<tr>
<th>RG - Region</th>
<th>RG - Ctry</th>
<th>RG - ID</th>
<th>SHIP_REF - Order_ID</th>
<th>SHIP_REF - Wt</th>
<th>SHIP_REF - Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utah</td>
<td>C</td>
<td>10</td>
<td>2007019003</td>
<td>10</td>
<td>KILO</td>
</tr>
<tr>
<td>Durango</td>
<td>D</td>
<td>73</td>
<td>2007010204</td>
<td>10</td>
<td>KILO</td>
</tr>
<tr>
<td>Hunan</td>
<td>F</td>
<td>80</td>
<td>20070408011</td>
<td>25</td>
<td>KILO</td>
</tr>
<tr>
<td>Punjab</td>
<td>B</td>
<td>312</td>
<td>20070329026</td>
<td>25</td>
<td>KILO</td>
</tr>
<tr>
<td>Delhi</td>
<td>B</td>
<td>258</td>
<td>20070327004</td>
<td>120</td>
<td>KILO</td>
</tr>
<tr>
<td>Texas</td>
<td>C</td>
<td>25</td>
<td>20070114001</td>
<td>25</td>
<td>KILO</td>
</tr>
<tr>
<td>Ottowa</td>
<td>A</td>
<td>60</td>
<td>20070222008</td>
<td>80</td>
<td>KILO</td>
</tr>
<tr>
<td>New York</td>
<td>C</td>
<td>213</td>
<td>20070314012</td>
<td>60</td>
<td>KILO</td>
</tr>
<tr>
<td>Taipei</td>
<td>F</td>
<td>120</td>
<td>20070211001</td>
<td>60</td>
<td>KILO</td>
</tr>
</tbody>
</table>

The actual data reveals that while RG.ID and SHIP_REF.WT do have a high statistical match that indicates a possible PF Key relationship, the relationship may not be valid; the WT column appears to be the shipping weight, not a region ID. If you are familiar with the SHIP_REF table you may determine right now that the relationship is not valid, or you may need to investigate the SHIP_REF table to find out for sure what the WT column represents.

Discover PF Keys

About this task

Before discovering primary-foreign keys, you must perform Column Analysis.

Note: If the source and target data sets contain only database tables (no text files), and all of the database tables contain correct primary-foreign key information in metadata, and the JDBC driver supports metadata import, you do not need to perform this step.
To skip this step, click the **PF Keys** tab. The *PF Keys* window opens displaying the tables in each data source. If the metadata contained PF Key information, it is displayed here. No column matches or PF Keys are displayed for tables that do not contain that information in metadata, or for text files.

**Procedure**

1. Click **Run Next Steps...** in the *Column Analysis* window. The *Processing Options* dialog appears with **PF Keys** selected.

2. If desired, drag down the **Steps** arrow to execute additional steps at the same time.
3. Click **Sub Steps**.
4. Check the appropriate sub steps. You can customize the sub steps for all selected steps.
   - To update the PFK statistics, check **PF Keys Refresh Stats**. This does not discover new PF Keys, but refreshes existing PF Keys after you have changed source data or imported a table or CWM file containing foreign keys. You cannot run any other PF Keys sub steps when this option is selected.
   - To discover or rediscover column matches, check **Column Matches**. When this action is selected, you can also choose to perform column classification using value matching.
   - To discover or rediscover PF Keys, check **PF Keys**. All unapproved discovered PF Keys are regenerated. Approved and user-defined PF Keys are not changed.
**Note:** If the source or target data sets have been modified (for example, by changing a sample set), do not run the **PF Keys** task until you have refreshed the statistics for all column matches using the procedure described in [Show All Column Matches](#).

- To discover or rediscover table and relationship classifications, check [Classifications](#).

5. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

**Note:** Modifying task options is an advanced activity. Options are described in [Reference of Task Options](#). Modifying options and creating new option sets is described in [Option Sets](#).

6. Click **Run** to execute the selected processing. If the procedure does not complete correctly, see [Troubleshooting Task Execution](#).

**Note:** Although the PF Keys window is displayed immediately, it may take some time for the actual data processing to complete. Check the [Activity Viewer](#) for the status of the task.

### Discovering composite PF keys on large tables

Discovery can discover composite PF key relationships very quickly even on tables with greater than 2000 rows.

### About this task

Discovery can use a sample of the relevant data from the primary sample data of each table in the data set to optimize the discovery of composite primary and foreign key relationships. The data in both the sampled discovery and the full composite key discovery algorithms are from the staging database. With some changes in the Process Options, Discovery returns PF keys for your large tables approximately ten times faster than if you used the full composite key discovery. If your tables contain a large number of rows, such as greater than 2000 rows, follow these steps to reduce the time it takes to discover composite PF Keys:

### Procedure

1. **Click Run Next Steps** in the Column Analysis window. The [Processing Options Dialog](#) opens with **PF Keys** selected. Either edit or create a new set of processing options to take advantage of the data sampling.

2. In the Edit Options window or the New Options window, in the **Step** drop-down list, select **PF Keys**.

3. Expand the **Generate PFKeys** section and change the following values to **True**:
   - Discover Composite Primary-Foreign Keys
   - Enable sampling algorithm to generate Composite PFKeys
4. Click OK to close the options window and begin the discovery process.

Results

Discovery returns PF Keys for your large tables approximately ten times faster than if you used full composite key discovery. This discovery uses the same statistics thresholds as that of the full composite key discovery.

Note:

Because this is a sampling, there is a possibility that some relationships might not be discovered, especially if the original relationships are relatively weak, such as a low hit rate.

Re-Run PF Keys

About this task

The Re-run Step button allows you to repeat one or more of the separate PF Keys discovery actions.
Use **Re-run Step** in the PF Keys window if:

- You want to repeat PF Key discovery using different option values.
- You want to repeat only Column Matches, PF Keys, or Classification discovery.
- You imported a table that contains foreign keys, or used the `Import CWM Files as Physical or Logical Tables` procedure with a CWM file containing foreign keys, and now you need to generate PFK statistics for the imported foreign keys.
- You ran the PF Keys task previously, made some changes in this screen, and now want Discovery to discover PF Keys again.
- You ran the PF Keys task previously and then changed the source data (by reloading tables, reimporting tables, or changing the data profile in Column Analysis), and now want to update the PF Key statistics without re-running PF Key discovery.
- You clicked directly on the PF Keys tab after Column Analysis, instead of using **Run Next Steps** from the Column Analysis window.

**Procedure**

1. Click **Re-Run Step** in the PF Keys window. The **Processing Options Dialog** appears with the additional options.
2. Check the appropriate options:
   - To update the PFK statistics after changing source data, or after importing a table or CWM file that contains foreign keys, check **PF Keys Refresh Stats**. This refreshes existing PF Keys and does not discover new ones. This action must be run by itself, it cannot be run with any other PF Key actions.
   - To discover or rediscover column matches, check **Column Matches**.
To discover or rediscover PF Keys, check PF Keys. All unapproved discovered PF Keys are regenerated. Approved and user-defined PF Keys are not changed.

**Note:** If you only need to run PF Keys, and the source or target data sets have been modified (for example, by changing a sample set), do not run this task until you have refreshed the statistics for all column matches using the procedure described in Show All Column Matches.

To discover or rediscover table and relationship classifications, check Classifications.

3. Verify that the task Options are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

**Note:** Modifying task options is an advanced activity. Options are described in Reference of Task Options. Modifying options and creating new option sets is described in Option Sets.

4. Click Run to re-run the selected PF Keys tasks.

**Note:** Although the next window is displayed immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.

**Define PF Keys**

You can manually define (add) PF Keys between related or unrelated tables, as well as edit and delete any PF Key.

**Note:** You cannot define a self join.

If two tables already contain columns with a primary-foreign key relationship, you also can create a new PF Key relationship between the tables in the opposite direction.

There are several ways to define a PF Key.

- Define a PF Key in the PF Keys List
- Define a PFK in the PF Keys Diagram
- Define a PFK in the PF Keys Grid

**Define a PF Key in the PF Keys List**

**About this task**

To define a PF Key in the PF Keys List

**Procedure**

1. Click a primary table to select it in the Connected Tables or Unconnected Tables list.
2. Click the + icon to display the **Pick a Foreign Table** dialog.
3. In the dialog, click a table to select it. The dialog lists only tables in the data set that do not already have a foreign key relationship with the selected primary table.

4. Click **Add**. The **Pick a Foreign Table** dialog closes.
   - If a column match already exists between the two tables, the **Promote Column Matches** dialog appears. Proceed as follows:
     a. Select one or more column matches and click **OK**. A new solid line connects the tables in the diagram, with the arrow pointing to the primary table. The selected column match also appears as a new PF Key entry in the PF Keys grid.
     b. Click **Refresh Statistics** to populate the new PF Key statistics. You can now start working with the new PF Key.
• If no column match exists between the two tables, a new solid line connects the tables in the diagram, with the arrow pointing to the primary table. A new PF Key entry also appears in the PF Keys grid identifying the tables but not the columns.

Define the columns in the new PF Key as described starting with Step 5 of the Define a PFK in the PF Keys Diagram procedure. After finishing, you can start working with the new PF Key.

**Define a PFK in the PF Keys Diagram Procedure**

1. Select **Foreign Key** from the **Arrow Mode** dropdown menu.

2. In the diagram area of the window, click once in the table containing the foreign key column. The actual column does not need to be visible.
Note: If there are many tables in the diagram, you can now select **Selected Objects** from the **Display Mode** dropdown menu. All non-related tables disappear from the diagram. Now click on the name of the table with the primary key in the **PF Keys List**, drag it into the diagram, and continue with **Step 3**.

3. Click and hold the mouse in the white area of the table with the foreign key, then drag to the white area of the table with the primary key.
4. Release the mouse in the white area of the table with the foreign key. The PF Key is added to both tables in the Connected Tables list, and an entry is added in the Foreign Keys grid.

A new solid line connects the tables in the diagram, with the arrow pointing to the table with the primary key.
5. In the PF Keys grid, click the Classification dropdown menu and select the correct PF Key Relationship Classification for the relationship.

6. Click the dropdown menu for the PF Key SQL expression. Specify the correct primary and foreign key columns using The Expression Editor. The format of a PF Key expression is:

   <Predicate 1> AND <Predicate 2> AND ... AND <Predicate n>

   - Predicates are <Left Expression> <operation> <Right Expression>
   - Allowed operations are =, (+)=, !=, >, < >, <=, LIKE, NOT LIKE
• There must be at least one column reference on each side, meaning a comparison to a constant is not allowed. For example, P.C1=5 is not a valid PF Key.

• ‘left expression’ and ‘right expression’ refer to the columns, functions or constants.

• NULL is not allowed. For example, A IS NULL is not valid.

• Do not mix references to the primary or foreign column in the same predicate. For example, P.C1 + F.C3 = F.P4 is not valid.

7. Click the Refresh Statistics icon.

Results

The PF Keys grid is updated with statistics for the new PF Key, and the new PF Key is saved in the staging data source.
Define a PFK in the PF Keys Grid

About this task

When a PF Key is selected in the diagram, you can add another PF Key expression to the existing expression(s) in the PF Keys grid. (The arrow points to the table with the primary key.) The new PF Key has the same direction as the existing PF Keys and can be between any two columns in the related tables.

Note: To create a PF Key between two unrelated tables, or to create a PF Key between related tables in the opposite direction, follow the instructions in Define a PFK in the PF Keys Diagram.

Procedure

1. Select a PF Key in the PF Keys Diagram or the Connected Tables list. All discovered PF Keys between the two tables in the same direction are listed as expressions in the PF Keys grid.

2. In the PF Keys grid, click the + icon. A new expression containing the table names is added to the grid.
3. Click the dropdown menu for the PF Key SQL expression. Specify the correct primary and foreign key columns using The Expression Editor.

4. Click the Refresh Statistics icon.

**Results**

The PF Keys grid is updated with statistics for the new PF Key, and the new PF Key is saved in the staging data source.

To make the new PF Key the Primary PF Key:
1. Select the radio button in front of the PF Key name. The green dot indicates the primary PF Key.
2. Click the **Classification** dropdown menu and select the correct **PF Key** for the relationship.

---

**Edit a PF Key**

**About this task**

Edit the primary PF Key or an alternative by clicking in its **Expression** field. The **Expression Editor** opens, allowing you to modify the expression.

**Note:** Always **Refresh Statistics** after editing a primary or alternative PF Key.

If a PF Key is approved, modifying it will change it to unapproved. To keep the changes if you **Re-Run PF Keys**, re-approve the PF Key after you are finished editing it.

**Delete a PF Key**

There are two ways to delete a PF Key:

- **Delete a PF Key in the PF Keys Grid**
- **Delete a PF Key in the PF Keys Diagram**

**Delete a PF Key in the PF Keys Grid:**

**About this task**

To delete a PF Key in the PF Keys grid, highlight the PF Key row in the **PF Keys Grid - Foreign Keys** and click the **Delete** icon.
Delete a PF Key in the PF Keys Diagram:
About this task

Note: Deleting a PF Key in this manner deletes all relationships between the two tables, including all primary PF Keys, non-primary PF Keys, and column matches.

You can delete a PF Key in the PF Keys diagram by selecting the connecting line, right-clicking the mouse, and selecting Delete from the dropdown menu.

In the following example, deleting the selected relationship in the PF Keys diagram deletes the primary PF Key, the second PF Key listed in the grid, and the two column matches between the tables (not shown).

Define Column Matches
There are two ways to add a column match.
- Define a Column Match in the PF Keys Diagram
- Define a Column Match in the PF Keys Grid
Define a Column Match in the PF Keys Diagram
Procedure

1. Select Column Match from the Arrow Mode dropdown menu.

2. Click in one of the tables to select it. The actual related columns do not need to be visible.
3. Click and hold the mouse in the white area of the selected table, then drag to the white area of the other table in the column match.
4. Release the mouse in the white area of the unrelated table.
   The column match is temporarily added to both tables in the Connected Tables list, and a temporary entry is added in the Column Matches grid. A new dashed line connects the tables in the diagram.

   **Note:** If you click anywhere except in the Column Matches grid, the temporary column match will be cancelled.

5. In the Column Matches grid, click the **Choose a column** dropdown menu for the first table and select the appropriate column.
6. Click the Choose a column dropdown menu for the second table and select the appropriate column in the second table.

7. Click the Refresh Statistics icon.
Results

The Column Matches grid updates with statistics for the new column match, and the column match is saved in the staging data source.

Define a Column Match in the PF Keys Grid
You can add a column match from the Column Matches grid.

About this task

Note: To create a column match between two unrelated tables, follow the instructions in Define a Column Match in the PF Keys Diagram.

When a column match is selected in the diagram, you can add another column match expression to the existing expression(s) in the column matches grid. The new column match can be between any two columns in the related tables.

Procedure
1. Select a column match in the PF Keys Diagram or the Connected Tables list. All discovered column matches between the two tables are listed as expressions in the column matches grid.
2. In the column matches grid, click the + icon. A new, empty expression is added to the grid.

3. Click the dropdown menu for the first table and select a column.
4. Click the dropdown menu for the second table and select a column there also.

5. Click the **Refresh Statistics** icon.
Results

<table>
<thead>
<tr>
<th>HQ_ADDRESSES</th>
<th>HQ_SALARIES</th>
<th>Row Hit Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>HQ_ADDRESSES</td>
<td>HQ_SALARIES</td>
<td>94% (30/32)</td>
</tr>
<tr>
<td>EMPLOYEE_ID</td>
<td>EMPLOYEE_ID</td>
<td>100% (28)</td>
</tr>
<tr>
<td>AREA_CODE</td>
<td>BONUS</td>
<td></td>
</tr>
</tbody>
</table>

The column matches grid is updated with statistics for the new column match, and the new column match is saved in the staging data source.

**Edit a Column Match**

**About this task**

Edit a column match by clicking in the column name and selecting a different column.

**Note:** Always [Refresh Statistics] after editing a column match.

**Promote a Column Match**

You can change a column match into a PF Key by clicking the **Promote Column Match** icon. In the popup window, select the appropriate column match to be promoted and then select the primary table in the dropdown menu, then click **OK**. The relationship is moved from the Column Matches tab to the PF Keys tab.

**Manage Column Matches**

When a data set contains 10 or more tables, IBM InfoSphere Discovery does not automatically load and display all of the column matches in the PF Keys diagram or the PF Keys grid. You can manually load and display the column matches in this case.

**About this task**

Projects can contain ten of thousands of matches. The Manage Column Matches task lets you view and manage the subset of matches of current interest. For large projects this *divide and conquer* or *focused exploration* approach to incrementally review and refine discovered results is invaluable. When a data set contains 10 (which is the default and can be changed by editing the App.config file) or more tables, InfoSphere Discovery does not automatically load and display all of the column matches upon visiting the PF Keys page. You can selectively load and display the column matches in this case.

You can manually load and display all of the column matches between two tables by selecting a connecting PF Key line in the diagram and clicking the **Column**
Matches tab in the PF Keys grid. Alternatively, you can select a more specific set of column matches by using the Manage Column Matches task.

The total number of column matches currently loaded into memory is shown in the lower left border of the PF Keys screen.

Note: The first time you load column matches can take several minutes. Subsequent column matches are displayed more quickly.

Procedure
1. Click the Show All PF Keys dropdown menu in the PF Keys diagram and select Manage Column Matches. The Manage Column Matches screen appears, displaying all of the columns in the current data set, or alternatively all the columns in the data set grouped by column name in the top pane. Column name grouping is useful since often columns with the same name have column matches.

2. Display the appropriate view (either Logical Tables or Column Name Groups).
   - View: Logical Tables displays all logical tables in the data set. You might be interested in viewing and managing the set of column matches that intersect a known set of tables in the data set.
   - View: Column Name Groups displays every unique column name in the data set. The Tables with Column column indicates how many tables each column name appears in.

3. Optional: To filter the results based on selectivity or row hit rate, click the Load Matches dropdown menu and select Set Matches Load Criteria. Column matches that are PF Key candidates often have a high selectively and high row hit rate. If you do not specify a minimum selectively or row hit rate in the load criteria, it will default to the values in the active option set used for PF Key discovery.

4. Select one or more rows in the top grid.

5. Click Load Matches. All column matches intersecting the selected tables or columns of a minimum quality are loaded into memory and listed in the lower pane. Once these matches are loaded, useful match count statistics are available. Many of these columns are hidden by default. These counts are a measure of how well a set of tables or columns are related.

6. Optional: You can load additional column matches at any time. Changing the thresholds in Set Matches Load Criteria will result in all column matches being unloaded. The reason for this is to maintain a set of matches in memory that have a common minimum selectively and row hit rate. It would be confusing to analyze if set of loaded matches with varying match quality (the minimums) since the absence of a match could either mean it does not exist or it has not been loaded.

   Important: If you use the delete tool to remove a column match from the lower pane, the column match is permanently deleted. You will need to rediscover column matches to see that column match again or you can close the project without saving it.

7. Click Close. The loaded column matches are displayed in the PF Keys diagram and in the PF Keys grid.
Review PF Keys

About this task

In the PF Keys window, examine the discovered primary-foreign key relationships and column matches for the source and the target data sets.

The Row Hit Rate and Value Hit Rate statistics for column matches and PF Keys reveal how closely the data matches in each column pair. These statistics do not change unless the source data changes. However, changing the values of the task options may raise or lower the number of column matches and PF Keys discovered.

The main activities to be performed in the PF Keys window are:

- Delete incorrect or unacceptable PF Keys.
- As needed, modify existing PF Keys or create new ones to obtain the highest hit rates.
- As needed, add columns to the Skip Columns list to reduce subsequent processing time.
- Verify that table classifications are correct.
- Verify that PF Key classifications are correct.

To accomplish these goals, you may need to generate reports, meet with subject matter experts, or research the data outside of Discovery.

One way of working in this screen is described below. The order in which you perform steps, and which steps you perform, can vary widely while still yielding good results. You can use the Approval Check boxes to keep track of the PF Keys and tables you have verified.

For both the source and the target data sets:

Procedure

1. [Zoom in and out on the diagram as needed to locate and identify all of the tables.]
2. Rearrange the PF Keys Diagram as needed to focus on the first set of related tables you will work on.
3. Verify that each Table Classification in the set of tables is correct.
4. Verify that the PF Key Classification for each PF Key is correct.
5. Verify that all of the Unconnected Tables are classified correctly.
6. Click on each PF Key relationship in the PF Keys Diagram or in the Connected Tables list to display its expression(s) in the PF Keys Grid - Foreign Keys.
7. Verify that each PF Key relationship is valid and of high quality.
   - If there is more than one PF Key expression for the relationship, compare each expression’s column names and statistics to make sure the primary expression is the best one. Change the Primary PF Key if necessary.
   - Use Foreign Keys Data Preview – Show Hits, Show Misses and PF Keys Foreign Keys Data Preview – Show Duplicates as necessary to examine the data.
   - Use Show All PF Keys, Show All Column Matches as necessary to compare all discovered relationships in one list.
8. If you know that a relationship exists between two columns in the data set, but Discovery did not discover a column match or PF Key for it, manually Define PF Keys.

9. As needed, Edit a PF Key to improve its statistics.

10. If needed, review the PF Keys Grid - Column Matches. It is generally not necessary to review column matches unless Discovery did not discover any PF Keys for a particular column pair. In that case, you may want to examine the column match statistics for that pair to see if a valid relationship may exist.

Results

When all PF Keys and unclassified tables in both the source and the target data sets have been reviewed, you are ready to Discover Data Objects.

The PF Keys Window

As soon as you add a table or text file to a data set, it is shown in the Unconnected Tables list, even before you run PF Key discovery.

After PF Key discovery, the Connected Tables list shows all tables in the data set with relationships to each other. The discovered PF Keys and column matches for those tables are also listed. The PF Keys Diagram displays a graphical representation of the column matches and PF Keys for each data set.

Note: If the data set contains a large number of tables, the diagram opens with the Display Mode set to Selected Objects, and contains only the tables related to the object currently selected in the PF Keys List.

Common Window Elements describes the menus, tabs, and buttons (like Validate, Step and Activity Viewer) that are common to all windows.
PF Keys Data Set Tabs

About this task

Source PF Keys and target PF Keys are displayed on separate tabs. Clicking a data set tab displays the Connected Tables and Unconnected Tables in that data set.

The source data set and target data set tabs use the names you provided in the Data Sets screen. If you did not specify names, the default names are used.

PF Keys List

The PF Keys list shows logical tables grouped by Connected or Unconnected, for each data set. Every table (database tables and text files) in the data set is shown in this list, either in Connected Tables or in Unconnected Tables.

You can expand and collapse the lists to view the contents more easily. Doubleclick on the list name, or click the + and - icons, to expand and collapse each list.
PF Keys List Icons:

PF Key relationships are indicated by an arrow (>). The arrow points to the primary table, so the foreign table is first in the pair and the primary table is second.

Column matches are indicated by a colon (:). The two tables in the column match are listed in alphabetical order.

Connected Tables:

This list shows all tables in the data set that contain a column with a relationship to a column in another table in the data set. Under each table is a list of all other
tables with a PF key relationship or a column match with this table. Each PF Key and column match appears twice in the Connected Tables list: PF Keys appear once under the primary table and again under the foreign table, and column matches appear once under each table in the column match.

In the example below, the first group shows that a foreign key in HQ_ADDRESSES references a primary key in HQ_EMP. Also, at least one column in HQ_ADDRESSES has a column match relationship to a column in HQ_SALARIES. The HQ_ADDRESSES->HQ_EMP relationship is repeated under the HQ_EMP table, and the HQ_ADDRESSES:HQ_SALARIES relationship is repeated in the list of relationships for HQ_SALARIES.

Unconnected Tables:

The tables in this list do not contain any column matches or PF Key relationships with any tables in this data set.

Discovery classifies unconnected tables as root entity tables. In the Data Objects step, these tables are generated as Multiple-table and Single-table Data Objects

Approval Checkboxes:
Approving a PF Key or table is a convenient way to indicate or keep track of items that have been reviewed or verified, are considered acceptable, etc.

**Note:** Relationship approval is discussed under [Classification Approved](#). Column matches cannot be approved.

The approval status of a PF Key can affect how Discovery treats the PF Key if you re-run PF Keys discovery. For more information, see [Re-Run PF Keys](#).

**Approved PF Keys:**

**About this task**

Checking a PF Key’s approval checkbox indicates that you are approving the primary PF Key shown in the [PF Keys Grid - Foreign Keys](#). When you check (or uncheck) the box, the corresponding relationship under the other table’s entry in the **Connected Tables** list is automatically updated to match, along with the [Approved](#) checkbox in the Foreign Keys Grid Tools.

If any changes are made to an approved PF Key, such as changing the classification or modifying the expression, the PF Key becomes unapproved and its approval checkbox is cleared.

---

![Diagram](https://via.placeholder.com/150)

**Approving one PF Key equation automatically approves the corresponding equation listed under the other table**

---

**Approved Tables:**

Tables are approved independently of PF Keys and column matches; clicking the approval checkbox of a table does not affect the approval status of the PF Keys or column matches under it.
If you re-run classification discovery, an approved table’s classification will not change.

PF Keys Diagram

See the Diagram Pane description for general information about working in diagrams, including:

- Colors of Source and Target Data Sets
- Selecting Tables and Connections
- Columns in the Diagrams
- Primary Key icon
- Zoom
- Changing the Table Size
- Moving Tables
- Moving Connections

Viewing the Objects

About this task

The PF Keys diagram displays all column matches, PF Keys, connected and unconnected tables in the data set at the same time. You can zoom out to see all objects at once or zoom in on a particular area. Selecting a table or connection in the Connected Tables or Unconnected Tables list shifts the diagram to display that item.

You can change the kinds of items displayed by selecting a different Display Mode.

Lines and Arrowheads

In the PF Keys diagram, PF Keys are shown as solid lines with an arrow pointing to the table containing the primary key. The arrowhead shape indicates the PF Key Classification.

Column matches are shown as dashed lines with a diamond on each end.

Note: Column Matches do not have a classification.
If a number is shown on the PF Key line between two tables, one or more candidate keys are marked for those two tables. See “PF Keys grid columns” on page 334.

Table Classification
About this task

Each table has a classification type, shown above the table’s title in the diagram.

Tables are classified using either Operational or Data Warehouse terminology, depending on how the data set was classified:
Most tables in an Operational data set are classified as either a root entity or a child entity.

Most tables in a Data Warehouse data set are a fact or a dimension.

For more information, see [About Classification](#).

You can change a table's classification by clicking on the classification type in the diagram and selecting another type from the dropdown list. Changing the table classification causes a different type of data object to be discovered. See [About Classification](#).

**PF Key Classification**

**About this task**

PF Key classifications are viewed by right-clicking on the classification line.

**Note:** Column Matches do not have a classification.

**PF Key Relationship Classifications** are viewed by selecting and then right-clicking on the connecting line.

The arrow points to the table containing the primary key. The arrowhead also indicates whether the PF Key relationship classification is Parent-Child, Reference, or None.
The PF Key classification is also shown in the grid below the diagram. You can change the classification by selecting the PF Key in the diagram or in the Connected Tables List, then clicking on the Classification menu and selecting the appropriate type from the list. You can check the Classification Approved checkbox when the classification is correct.

Working With a Large Diagram
When you rearrange table icons and relationship lines in a PF Keys diagram that contains many tables and relationships, it can take a long time to load the tables or refresh the view after rearranging the diagram. The diagram may also be too large to see at one time.
About this task

Options for working with a diagram containing many tables include the following:

- Search in the PF Keys list for a specific table, select it in the diagram, and then change the Display Mode to a more restrictive filter, such as PF Keys Only or Selected Objects’ PF Key.
- In the Show All Column Matches list, delete the weaker column matches and then Re-Run PF Keys with only the PF Keys option selected. The weaker PF keys will be dropped during rediscovery.

PF Keys Diagram Pane Tools

The following tools are available in the diagram pane of the PF Keys window:

- Arrow Mode
- Skip Columns
- Display Mode
- Refresh Display
- Show All PF Keys, Show All Column Matches
- Reload Matches
- Zoom

Arrow Mode:

This option allows you to click and drag to manually Define a PFK in the PF Keys Diagram or Define a Column Match in the PF Keys Diagram between unrelated tables in a data set. Also, if two tables already contain columns with a primary-foreign key relationship, you can create a new PF Key relationship between the tables in the opposite direction.

Skip Columns:

About this task

This option allows you to specify certain columns in the data set for Discovery to ignore when rerunning the PF Keys step. This can reduce processing time by eliminating columns that you know have no relationship to other columns.

All columns in the data set that match an entry in this list are ignored. You can also enter an item using the format TABLE.COLUMN to ignore a column in a specific table.
**Note:** Discovery does not validate the entries in this list.

Discovery will only ignore columns that exactly and completely match entries in this list. If you enter a partial string here, columns containing that string will *not* be ignored.

To ignore a column when generating PF Keys:

**Procedure**

1. Click the **Skip Columns** button.

2. In the **Skip Columns** popup window, click in the entry field.

3. Type a column name in one of the following formats:
• Column name only. Discovery will ignore all columns in the data set that exactly match this name.
• In the format TABLE.COLUMN. Discovery will ignore only this table-column name.

4. To enter another column, click in the Column Name header or press Enter. The new column is created and another blank row appears.

   Enter as many columns as necessary. You can Sort Rows by Column and Auto Filter Row by clicking the icons in the column header.

   To delete an entry, highlight the text in the entry and right-click, then select Delete.
5. When you are done, click **OK**. Discovery saves the listed columns and will ignore them when discovering PF Keys.

**Display Mode:**

**About this task**

Changing the display mode allows you to focus on a smaller part of a large, complex diagram.

The available display modes are listed below. An example after each mode shows how the same data set diagram would appear using that mode.

**Note:** Depending on the table currently selected, the screen layout may change when you choose a different view.

- **View All**: displays all column relationships in the data set.
**PF Keys Only**: displays only the PF Keys in the data set. Dashed lines indicating column matches, and any tables related only by column matches, are not shown.

**Selected Objects**: Displays only the tables, PF Keys, and column matches that are directly related to the object (table or connection) selected in the PF Keys List.
• **Selected Objects' PF Key**: Displays only the tables and PF Keys that are directly related to the table or connection currently selected in the PF Keys list.

**Refresh Display:**

**About this task**

This button is active when you choose **Selected objects** or **Selected objects' PF Keys** in the **Display Mode**. Click it to rearrange the displayed items to fit on the screen.
Show All PF Keys, Show All Column Matches:

This option displays all PF Keys or all column matches in the data set, regardless of the Display Mode. The relationships are shown in a popup window.

- **Show All PF Keys**
- **Show All Column Matches**

**Show All PF Keys:**

The **Show All PF Keys** window is shown below.

**About this task**

The standard tools for [Grid Panes](#) are available for viewing and manipulating the data in the grid.

The **Approve PF Key** dropdown menu allows you to perform actions on one or more PF Keys at once.

Select one or more PF Keys in this window using shift-click or control-click to highlight the desired rows, then select the appropriate action to apply to all selected rows:

- **Approve PF Key**
- **Approve Classification** of the selected PF Keys
- **Drop Classification** of the selected PF Keys

In addition, you can click the red X icon to [Delete](#) all selected PF Keys.
The columns in the grid are:

- **Alternative Count**
- Approve PF Key: mark the PF Key as approved
- Approve Classification: mark the classification as approved
- **Classification**
- **Origin**
- **Expression (Show All PF Keys)**
- **Table**
  - **Primary**: the table containing the primary key
  - **Foreign**: the table containing the foreign key
- **Row Hit Rate**

**Show All Column Matches**:

The **Show All Column Matches** window is shown below.

**About this task**

The standard tools for Grid Panes are available for viewing and manipulating the data in the grid. You can refresh statistics for all column matches by pressing CTRL-a to select all rows in the list and then clicking **Refresh Statistics**.

In addition, the **Data Types, Mixed** column indicates whether the column contains both numeric and nonnumeric data types. When a column has mixed data types, preview the data to determine whether the results are accurate.

**Sort by Rate, Sort by Count**:

**About this task**

In addition, you can choose the sort order for columns displaying both percentage and fraction statistics. Select the desired sort method in the dropdown menu and then click in the header of a column to sort the rows according to the results in that column.

- **Sort by Rate** sorts the column by the percentage.
- **Sort by Count** sorts the column by the top number in the fraction.

The following example shows the same rows sorted by rate and by count.
Reload Matches:

Column matches are discovered based on the current PF Key row hit rate and selectivity option values. However, Discovery often finds many more matches that are below these thresholds, and which are therefore not displayed in the results.

About this task

It can be useful sometimes to review the column matches in the current data set that do not meet the minimum thresholds. For example, if the sample set is small, you may want to promote column matches that you know are valid for the full data set. Or, if you are going to run composite foreign keys, you can delete low-quality column matches so they do not participate in the discovery.

One way to view the column matches that do not meet the original thresholds is to re-run the Column Matches task with lower option thresholds. However, you can also simply reload the list of column matches using lower thresholds. This is similar to lowering the filter to display additional column matches, without actually re-running the task.

Procedure

1. Display Show All Column Matches

2. Click Reload Matches. The Column Matches Reload Criteria dialog appears.
3. Change the **Row Hit Rate Threshold** and **Selectivity Threshold** to the desired values, then click **Reload**. The *Column Matches Reload Criteria* dialog closes.

**Results**

The *Show All Column Matches* window is repopulated. The reloaded list contains all previously-discovered column matches in which either the Left or Right Row Hit Rate, or the Left or Right Selectivity, meet or exceed the new thresholds.

**Note:** If the new criteria include a large number of column matches, it may take a long time to display the results.

**PF Keys Grid - Foreign Keys**

*When you select a PF Key relationship in the [Connected Tables](#) list or in the [PF Keys Diagram](#) the Foreign Keys grid tab is automatically displayed.*

**Note:** The PF Keys grids display information about PF Keys and column matches. Tables have no information that can be displayed in the PF Keys grids.

**Primary PF Key**

The Foreign Keys tab displays all discovered PF Keys between the two tables. Each PF Key is shown as a SQL expression. The statistically strongest PF Key is selected as Primary. It is indicated by a green button.
Foreign Keys Grid Tools

The following standard tools are available for viewing and manipulating the data in the Foreign Keys grid. Also see the Grid Panes description for common actions you can perform in a grid.

- Add a PF Key
- Delete a PF Key (changes it to a column match)
- Refresh Statistics after adding or modifying a PF Key
- Column Chooser
- Foreign Keys Data Preview – Show Hits, Show Misses
- PF Keys Foreign Keys Data Preview – Show Duplicates

In addition, you can approve a primary PF Key and its classification. An approved PF Key will not be rediscovered if you re-run the PF Keys step. Approval is a convenient way to indicate or keep track of items that have been reviewed or verified, are considered acceptable, etc.

Note: Alternative PF Keys (SQL expressions not marked as primary) cannot be approved.

Approved:

About this task

This marks the primary PF Key as approved. The corresponding Approval Checkboxes in the Connected Tables list are also updated. Making any changes to the primary PF Key, including making a different PF Key the primary, changes the status of all three checkboxes to unapproved (unchecked).
Classification Approved:

This marks the primary PF Key's classification as approved.

You can change the primary PF Key Relationship Classification by selecting the primary PF Key in the grid and clicking the Classification dropdown menu, then selecting a different classification. The options in the menu depend on the Data Set Classification.

PF Keys grid columns

The following common information is displayed in the Foreign Keys grid by default:

- SQL Expression of the discovered PF Keys
- Row Hit Rate
- Value Hit Rate
- Selectivity

Candidate Keys

Use the Column Chooser to display the Candidate Keys column. Candidate keys can include any of the alternative PF Keys. To mark an alternative key as a candidate key, click the checkbox in the Candidate Keys column.

When two or more PF Keys are marked as candidate keys, the PF Keys diagram indicates the number of candidate keys in the connecting line between the two tables. For example, when no PF Keys are marked as candidate keys, the PF Keys diagram shows an unbroken line between the two tables, indicating the tables have a PF Key relationship (the primary PF Key). If you mark two alternative PF Keys as candidate keys, the PF Keys diagram shows the number 3 on the connecting line between the two tables.

Note: A primary key is implicitly a candidate key by default and cannot be marked as a candidate key.

Candidate keys are exported along with the primary PF Key.

Foreign Keys Data Preview – Show Hits, Show Misses

About this task

When you select an expression in the PF Keys Grid - Foreign Keys and click Show Hits, the PF Keys Show Hits or Show Misses Data Preview displays the hits and misses for the discovered PF Keys for the selected expression.

- Show Hits displays the rows in the primary and foreign key columns that have hits in the other table. These are the rows in the Row Hit Rate statistic.
- Show Misses displays the rows in the primary and foreign key columns that do not have hits in the other column.

You can also Select Preview Criteria
PF Keys Foreign Keys Data Preview – Show Duplicates

About this task

Show Duplicates displays source, query and target data for the selected expression when the binding condition results in duplicated rows. It can be useful to examine the duplicates when either the source column or the target column Selectivity is less than 1.

To display the primary and foreign keys for a duplicated row, select the row in the Primary Duplicates pane and click Focus.

You can also Select Preview Criteria.
PF Keys Grid - Column Matches

About this task

When you select a column match in the Connected Tables list or in the PF Keys Diagram, the Column Matches grid tab is automatically displayed.

When a discovered PF Key is selected, you can also click the Column Matches tab to display the column matches that the PF Keys are based on. There are no column matches listed for manually created PF Keys.

Note: The PF Keys grids display information about PF Keys and column matches. Tables have no information that can be displayed in the PF Keys grids.

The Column Matches tab displays all column matches between two tables. In most cases, at least one of these column matches has been selected as a PF Key.

Column matches differ from PF Keys in the following ways:

- Column matches are non-directional. There is no concept of primary or foreign key in the Column Matches grid.
- There is no primary column match.
- You cannot approve column matches.
- Column matches do not have a classification.
Column Matches Grid Tools

The following standard tools are available for viewing and manipulating the data in the Column Matches grid. Also see the Grid Panes description for common actions you can perform in a grid.

- Add a column match
- Delete a column match (removes the relationship from the staging data source)
- Refresh Statistics after adding or modifying a column match
- Column Chooser
- Column Matches Show Hits, Show Misses

Column Matches Grid Columns

The following common information is displayed in the Column Matches grid by default:
- SQL Expression of the discovered column matches
- Row Hit Rate
- Value Hit Rate
- Selectivity

Column Matches Show Hits, Show Misses

About this task

When you select an expression in the PF Keys Grid - Column Matches and click Show Hits, the Column Matches Show Hits Data Preview displays all discovered column matches for the selected expression.

Show Misses displays the rows that are not included in the selected column match.

You can also Select Preview Criteria
Discover Data Objects

In this step, Discovery organizes related tables into structures called data objects, based on the primary-foreign keys. Data objects are represented in this step as diagrams and also as lists of tables.

Overview

Your task in the Data Objects window is to verify that the primary-foreign keys and classification types created data objects that are sensible and complete, as measured by the statistics in this window and your own knowledge of the data.

Once you review data objects, Discovery maps them to one another in the Target Matches step. Target Matches and Maps produce successful results only if the data objects correctly and accurately represent data relationships.

In addition to the data objects that Discovery discovered, you can Define Data Objects in this screen, and change data objects that Discovery discovered.

You can also entirely skip the process of having Discovery discover data objects and instead manually define all of them in this screen. Your knowledge of the data must be very good in order to manually define data objects.

What Is a Data Object?

A data object is a logical cluster of all tables in the data set that have one or more columns containing data related to the same business entity. Data objects are not
maps, but instead represent an object view of related tables. Grouping tables in this way allows Discovery to narrow the focus of analysis to only the tables that are known to be related.

For example, in the PF Keys step, Discovery may have discovered that some PF keys connected columns with information about employees, and other PF Keys connected columns with information about orders. In the Data Objects step, Discovery would create one data object containing only the employee-related tables and a separate data object with only the order-related tables. If one table contained information about employees and orders, it would be represented in both data objects.

The following diagram illustrates how data objects look in the context of a large database schema.

Why Use Data Objects?
Mapping tables as data objects is a much smaller and faster exercise than attempting to map all columns in a data set. The discovery is more focused and allows you to review the discovered mappings in a smaller context, without being overwhelmed by the large number of details typically contained in a large mapping.

You can also fine-tune the semantics of individual schemas by manipulating data objects, including adding or deleting tables, and modifying the column matches within the data object. Correctly defined data objects greatly improve Discovery’s ability to automatically discovery source-to-target mappings.

For these reasons, data objects are an excellent way to preview and refine the relationships within a data set before beginning to map target data to source data.

The major benefits of creating data objects before mapping source data to target data include:
• Rather than using a highly memory-intensive technique of trying to map every column in every table on the source side to every table and column on the target side, Discovery can perform the much simpler task of comparing the source and target data objects to each other, limiting the effort to mapping one set of related columns at a time.

• Data objects allow you to think about a large schema at the object level. This allows you to focus on correcting the join semantics in a business context.

• Data objects provide a meaningful context for thinking about source to target relationships.

**How Discovery Generates Data Objects**

Discovery discovers data objects based on PF keys. For this reason, it is very important to carefully verify the PF Keys in the previous step. If your project does not have PF Keys, Discovery cannot discover data objects.

The relationships between individual columns (and the resulting relationships between tables) were discovered in the PF Keys step. In [Review PF Keys] you verified and refined the discovered relationships and made sure the correct primary-foreign key was selected for each relationship, along with confirming that each [PF Key Relationship Classification] and [PF Key Table Classification] was correct.

**Impact of Classification Type on Data Objects:**

The exact structure of each data object depends on the classification type of each table and column, as well as the [Data Set Classification]. In most cases, tables classified as root entity tables become root tables (parents). Tables with foreign keys classified as child entity or reference tables usually become child tables in data objects.

**Multiple-table and Single-table Data Objects:**

Most data objects consist of two or more related tables. However, tables with no primary or foreign keys are also considered data objects. Discovery considers an unrelated table as the root table of its own data object, containing all of the data related to its own entity.

Because PF Keys and data object discovery is performed on each data set separately, a table with no relation to another table in its own data set may be related to a table in the other data set. Or, the table may be related to a table in its own data set in a way that is not yet obvious. These relationships will be discovered in subsequent processing steps.

**Data Object Naming Conventions**

Discovery names each automatically-generated data object based on its root table. The format is:

\[DO_root_table_name\]

Discovery also automatically generates a description for each discovered data object. You can change this description at any time in the [Notes] column.

For manually-defined data objects, you enter a name when you create the data object, and you can optionally add notes.
Names of Duplicate Data Objects:

If Discovery generates a duplicate data object (as described in Generate and Regenerate Data Object Options), Discovery gives the duplicate data object the same name as the original and appends an incrementing number, starting with 0. You can change this name after the Discover Data Objects task has completed.

Related Sample Sets

Data objects allow you to create related sample sets, which are an important tool in Discovery.

The tables in every data object are related by join conditions. When you define a sample set for a table and use that sample set to define sample sets for all of that table's related tables, you are creating related sample sets. You can select a root table or a child table as the starting table for related sample sets.

The full content of each table (Full Set sample set) is always used as the starting point when creating a related sample set, regardless of the sample set that is currently set as primary for each table. The join condition is applied to the full content of all tables in the data object, and the resulting rows are filtered using the sample set you select in Create a related sample set.

Discovery does not change the primary sample set after creating related sample sets. To use the related sample sets as primary, click on the Data Sets tab after you Preview Sample Set Contents, then click the related sample set's radio button to make it the primary. Then click back on the Data Objects tab and Discover Target Matches.

Example of Related Sample Sets:

For example, assume the table HQ_EMP is the root table of an existing data object containing two child tables, HQ_ADDRESSES and HQ_SALARIES. In the Data Sets tab, leave the Full Set as the primary sample set for HQ_EMP but also define a sample set called Current, defined by the Where clause STATUS='Current'.

To create related sample sets using the HQ_EMP data object, right-click on HQ_EMP in the Data Objects diagram and select the Current sample set, then click OK. Discovery generates a sample set for each child table and filters out all rows not related to the Current records in the root table. The new related sample sets are based on the Current sample set.

The new related sample sets are added to each child table's entry in the Data Sets window, but are not made primary. If you make these related sample sets the primary sample sets for each table and then run Maps discovery, Discovery focuses only on the data that is useful in mapping the tables to each other. By using related sample sets, you have reduced the amount of processing time needed and avoided the possibility of sample sets “missing” each other.

Note: When a data object contains an alias, the related sample set for that alias is added to the original table's entry in the Data Sets window.

Location of Related Sample Sets:
New related sample sets are immediately available in the Data Sets window. Each sample set appears under the child table’s entry. Related sample sets based on a table alias appear under the original table’s entry.

**Related Sample Set Naming Conventions:**

Related sample sets are automatically named using the following convention:

SS_for_<do-name>_<child-name>_<filter-sample-set>_<child-primary-sample-set>

You can change this name in the Data Sets window.

**Discover Data Objects**

**About this task**

Discovery creates data objects from all of the discovered or imported PF keys in each data set. You cannot select certain PF Keys to use, or select only the source or target data set, when generating data objects.

**Note:** Discovery does not use tables with a System classification when generating data objects. If a PF Key includes any System tables, Discovery will ignore them in this step. See [PF Key Table Classification](#).

To advance to the Data Objects window without performing the Data Object processing first, click the Data Objects tab. The Data Objects window opens with an empty Data Objects list and diagram pane. This allows you to [Define Data Objects](#) instead of having Discovery generate them.

**Procedure**

1. **Click Run Next Steps** in the PF Keys window. [The Processing Options Dialog](#) appears with Data Objects selected. The Processing Options Dialog for SDD projects is shown below. TD projects have a similar dialog. The Data Objects step has no sub steps.
2. (SDD projects only) If desired, drag down the Steps arrow to execute Overlaps in the same step. Configure the Overlaps sub steps.

3. Verify that the task Options are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   Note: Modifying task options is an advanced activity. Options are described in Reference of Task Options. Modifying options and creating new option sets is described in Option Sets.

4. Click Run to execute Data Objects processing. This is performed on all data sources (database tables and text files) in the data sets. If the procedure does not complete correctly, check the Troubleshooting Task Execution section.

   Note: Although the Data Objects window is displayed immediately, it may take some time for the actual results to show. Check the Activity Viewer for the status of the task.

**Generate and Regenerate Data Object Options**

Once Discover Data Objects has been run and data objects exist (or after you Define Data Objects), the Generate Unique Data Object and Regenerate Data Object options are important. These options are related to each other, and their settings affect existing data objects when re-running this step. The outcome also depends on whether you have used the Data Object Approval Checkboxes to approve a particular data object.
It is important to understand how these options affect existing data objects, because you can inadvertently drop a data object during this step. In other cases, you can create duplicate data objects or cause a data object to not be regenerated.

The following table describes the outcome on a data object in all scenarios: approved or unapproved, and with each possible combination of settings for the two options.

**Note:** For the purposes of data object generation, Discovery considers any *defined* data object as approved, along with any *discovered* data object that you have modified. This is true even if you have not checked the data object approval checkbox for that particular data object.

<table>
<thead>
<tr>
<th>When Generate Unique DO is...</th>
<th>and Regenerate DO is... for an existing DO that is...</th>
<th>when you re-run Generate Data Objects, that data object is...</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>Approved left as is</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>Approved left as is</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>Approved left as is, and a duplicate data object is generated</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>Approved left as is, and a duplicate data object is generated</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>Not approved dropped and regenerated</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>Not approved left as is</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>Not approved dropped and regenerated</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>Not approved left as is, and a duplicate data object is generated</td>
</tr>
</tbody>
</table>

To summarize the table, an approved data object is *never* dropped when you re-run Generate Data Objects. Depending on the *Regenerate Data Object* setting, a duplicate may be created for it.

Unapproved data objects are dropped when *Regenerate Data Object* is True, and kept when it is False. Depending on the *Generate Unique Data Object* setting, duplicate data objects may be created.

### Specifying Boundaries of Related Tables

The classification types of the data source, tables, and relationships determine the structure of the data object that Discovery discovers. In addition, the following two options allow you to specify the types of tables that Discovery includes as related tables when it generates data objects.

- **Data Object generation includes Reference tables**
- **Data Object generation includes attribute tables of reference tables**

**Note:** Cross-reference tables are always included as related tables in both data objects where its parent tables are root.

These options are available in the *Generate Data Objects* group in the Options menu.

The effects of these options on the following types of data sets is shown below.

- **Data Objects in an Operational Data Source**
• Fact Table Data Objects in a Data Warehouse Source
• Dimension and Reference Data Objects in a Data Warehouse Source

Data Objects in an Operational Data Source:

The following table describes the depth of data objects created from Operational Data Source root entity tables, depending on the settings of the options above.

Note: Cross-reference tables are always included as related tables in both data objects where its parent tables are root.

<table>
<thead>
<tr>
<th>When Include Reference Tables is...</th>
<th>...and Include Attribute Tables of Reference Tables is...</th>
<th>...the Data Object based on an Operational Data Source root entity table contains this:</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>All children of the root entity table, plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All levels of tables with a Reference relationship to this table, plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All levels of children of all tables with a Reference relationship to this table</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>The first level of tables with a Reference relationship to this table, plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All children of the root entity table</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>All children of the root entity table</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>All children of the root entity table</td>
</tr>
</tbody>
</table>

Fact Table Data Objects in a Data Warehouse Source:

The following table describes the depth of data objects created from Data Warehouse fact tables, depending on the settings of the options above.

Note: Cross-reference tables are always included as related tables in both data objects where its parent tables are root.

<table>
<thead>
<tr>
<th>When Include Reference Tables is...</th>
<th>...and Include Attribute Tables of Reference Tables is...</th>
<th>...the Data Object based on a Data Warehouse fact table contains this:</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>All direct child tables, plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All dimension tables of this fact table, plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All child tables linked to the dimension tables</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>All direct child tables, plus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All dimension tables of this fact table</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>All direct child tables of this fact table</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>All direct child tables of this fact table</td>
</tr>
</tbody>
</table>
Dimension and Reference Data Objects in a Data Warehouse Source:

The following table describes the depth of data objects created from Data Warehouse dimension and reference tables, depending on the option settings.

**Note:** Cross-reference tables are always included as related tables in both data objects where its parent tables are root.

<table>
<thead>
<tr>
<th>When Include Reference Tables is...</th>
<th>...and Include Attribute Tables of Reference Tables is...</th>
<th>...the Data Object based on a Data Warehouse dimension or reference table contains this:</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>All direct children, <strong>plus</strong> all levels of tables with a Reference relationship to this table, <strong>plus</strong> all child tables of Dimension or Reference tables</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>all direct children, <strong>plus</strong> one level of tables with a Reference relationship to this table</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>one level of children of this table</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>one level of children of this table</td>
</tr>
</tbody>
</table>

**Re-run Data Objects**

The **Re-run Step** button allows you to repeat the Data Objects discovery step as individual actions, while you are in the Data Objects window.

**About this task**

**Note:** Regenerating a data object may affect existing maps. If you have existing maps, rediscover them after performing any action that generates a new data object, including manually creating a new data object.
Use **Re-run Step** in the Data Objects window if:

- You clicked directly on the Data Objects tab to manually create data objects after PF Keys, instead of using **Run Next Steps** from the PF Keys window. Now you want Discovery to automatically discover data objects.
- You ran the Data Objects task previously and then created or modified some of the discovered data objects. Now you want Discovery to rediscover data objects based on your changes.
- You have made any changes to the data sets and need to repeat the Data Objects step to update the results.

**Procedure**

1. Click **Re-Run Step** in the Data Objects window. The **Processing Options Dialog** appears.
2. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   **Note:** Modifying task options is an advanced activity. Options are described in **Reference of Task Options**. Modifying options and creating new option sets is described in **Option Sets**.
3. Click **Run**. Discovery discovers Data Objects for all data sources (database tables and text files) in the data sets.
Note: Although the Data Objects window is displayed immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.

Define Data Objects

You can manually create (define) data objects. In defined data objects, a table can be used as a root table for as many data objects as necessary. (When you run the Discover Data Objects task, a table is used as a root table for only one data object.)

In addition, when using the Parent-Child Relationships option in the following procedure, the root table and the other tables and relationships do not need to be correctly classified in order to create the data object. For all other options, and for automatically-discovered data objects, the root table, other tables, and relationships do need to be correctly classified.

Define a Data Object

To add a data object:

About this task

Procedure
1. In the Data Objects List click +.

The Create New Data Object screen appears, listing all tables in the data set. If the table contained a description in metadata or you added a description when importing the table to the data set, that is also shown.
2. Enter a name for the new data object. You can also add a description, but this is not required.

**Note:** Using the default Discovery format for data object names may have implications if you **Re-run Data Objects**. For more information, see **Data Object Naming Conventions**.

3. In the **Tables** list, select a table to use as the root table.
4. Select the appropriate level of related tables and relationships to include in the data object. For more information, see Specifying Boundaries of Related Tables.

- **Direct Children** includes tables designated as child entities in the PF Key relationship.
- **Direct Children and References** includes tables designated as child entities in the PF Key relationship, along with one or both of the following:
  - *tables that have an Attribute relationship* with tables that have a Reference relationship to the selected root table
  - *tables that have a Reference relationship* with the selected root table
- **Parent Child Relationships** includes both parents and children of the selected root table, up to the selected number of levels. In this option, the root tables, other tables, and relationships do not need to be correctly classified.

5. Click OK. The new data object is added to the **Data Objects** list with all specified related tables.
Rename a Data Object

About this task

A data object's name and description can be changed at any time to make it more descriptive of the business entity it defines.

For example, a data object with the default name of DO_07112SW may define a data object containing sales information for the southwest region for 2007. In this case, it can be helpful to change the default data object name to a more meaningful one, such as DO_SALES_SW_2007.

Note: Changing a default data object name can change the discovery results if you [Re-run Data Objects](#). For more information, see Data Object Naming Conventions and Generate and Regenerate Data Object Options.

To rename a data object or change its description:

**Procedure**

1. Click on the data object name in the diagram. The Data Object Properties dialog appears.
2. Delete the existing data object name and type a unique new name. You do not need to keep the DO_ prefix.

3. Delete the existing description and type a new one. (This is an optional field.)

4. Click OK.

**Delete a Data Object**

**About this task**

Deleted data objects cannot be recovered. If you accidentally delete a data object, Re-run Data Objects or Define Data Objects.

To delete a data object, select the data object name in the Data Objects list and then click the X icon.
The data object is deleted from the staging data source.

**Add Child Tables**

**About this task**

You can manually add child tables to any table (root or child) in a discovered or manually created data object. If you add a child table that is already part of the data object, Discovery Studio adds an alias of the table to the data object. The alias has the same name as the original but is appended with an incrementing digit, starting with 0.

**Note:** Remember to Refresh Statistics after entering each join condition.

**Procedure**

1. Select a root or child table in the Data Objects Diagram.
2. Right-click on the selected table and choose Add Child Table... from the dropdown menu. The Add Table window appears.
3. In the Add Table window, review the listed tables. All tables in the data set are shown, regardless of their PF Key Table Classification.
   - **Reference to <table>** shows the relationship, if any, of each listed table to the table selected in the diagram.
   - **Reference by <table>** shows the relationship, if any, of the table selected in the diagram to each listed table.
     - If a field is blank, no relationship exists.
4. Select a table in the list and click OK.
The new table is added to the data object as a child, using the table classification assigned in the PF Keys window.

5. Select the new join in the diagram. Its expression is shown in the Data Objects Grid. If the new child table had an existing relationship with the selected table, the join expression is automatically created. Click Refresh Statistics to refresh the statistics, and you are finished.

If the table was not related or the join is incomplete, complete it as described in the next step.

6. Click in the SQL expression to display the Expression Editor then specify the join expression.
7. Click the Refresh Statistics icon to generate statistics for the new join.

Delete Child Tables
About this task

To delete a child table from a data object, select the table and then right-click on it. From the dropdown menu, select Delete.

The child table is deleted from the data object.
Create a Related Sample Set

About this task

If the sample set you want to use as a filter does not exist, you can Define a Sample Set in the Data Sets tab, then return to the Data Objects tab and continue with creating related sample sets. The sample set you created is immediately available for use. You do not need to re-run any processing steps before using the new sample set as a filter here.

To create a related sample set:

Procedure
1. Select a table in the data object.
2. Right-click on the table and select Create Related Sample Set. In this example, the selected table is a parent to two children, so two related sample sets will be created.
3. In the Choose sample set popup, select a sample set from the dropdown menu. The list contains all sample sets for this table in the Data Sets screen. The sample set you choose here will be used to drive the new related sample sets.

4. Click OK. Discovery creates the related sample sets. A progress dialog indicates when it is finished.

5. Click on the Data Sets tab to display the database connections and tables in this data set. A new sample set for each child appears under each child's entry in the list.
**Review Data Objects**

Once a data object has been created, you can edit it. You can Add Child Tables and Delete Child Tables and also modify the join conditions. However, you cannot delete the root table. If needed, you can also Define Data Objects.

**The Data Objects Window**

If you display this window before discovering Data Objects, the Data Objects list and the diagram are empty. Once data objects exist, the diagram displays the data object that is selected in the Data Objects List.

Common Window Elements describes the menus, tabs, and buttons (like Validate Step and Activity Viewer) that are common to all windows.
Data Objects Data Set Tabs

About this task

Each data set contains a set of data objects. Click a data set tab to display the data objects in that data set.

The source data set and target data set tabs use the names you provided in the Data Sets screen. If you did not specify names, the default names are used.

Data Objects List

After discovering Data Objects, the Data Objects list contains the logical tables in each discovered data object, grouped by root (parent) table and by data set.
About this task

Note: If you defined data objects manually, the list may not contain all tables in the data set. Tables not included in a Data Object will not be considered in subsequent discovery steps.

You can expand and collapse the lists to view the contents. Double click on the list name, or click the + or - icon, to expand and collapse each list.

The following screenshot shows a Data Objects list containing two data objects.

Data Object Root Tables

The root (parent) table is the first table listed in each data object, under the data object name.

When Discovery discovers data objects, each table in the data set is represented no more than once as a root table, either for Impact of Classification Type on Data Objects or as Multiple-table and Single-table Data Objects. The discovery task will not create two data objects that have the same root table. You can Define Data Objects if you want to use a table as a root table for a second data object.

Child tables are indented under the root tables. A table can appear as a child table in as many data objects as necessary.

When a join is selected in the Data Objects Diagram, its expression is shown in the Data Objects Grid.

Data Object Approval Checkboxes

Approving a data object or join is a convenient way to differentiate those that have been reviewed and verified as acceptable, from those that have not yet been reviewed. A data object's approval status has no impact on Target Matches or Maps data discovery.
Note: The approval status of a data object is important if you re-run the Data Objects step, since it determines whether Discovery will drop and regenerate the data object, create a duplicate, or take no action. For more information, see Re-run Data Objects.

Individual tables within a data object cannot be approved, but joins can be approved in the Data Objects Grid.

Data Objects Diagram

About this task

The Data Objects diagram displays data objects within each data set, based on PF Keys.

The source and target data sets each have their own list of data objects. To view the data objects in each data set, click the appropriate Data Objects Data Set Tabs.

The Data Objects diagram displays only the selected data object. To view a different data object, click on the data object name or child table in the Data Objects list.

You can rearrange child tables individually in the diagram. Moving the root table moves the data object as one unit.
See the Diagram Panes description for general information about working in diagrams, including:

- Colors of Source and Target Data Sets
- Selecting Tables and Connections
- Columns in the Diagrams
- Primary Key
- Zoom
- Changing the Table Size
- Moving Tables
- Moving Connections

Data Object Diagram Lines and Arrowheads
Joins are shown as solid lines with an arrow pointing to the child table.

---

### Mapping Studio - LOCALHOST - TestProject2

![Data Objects Diagram](image)

**Data Objects Grid**

**About this task**

When you select a join in the Data Objects Diagram, all discovered SQL expressions for that join are displayed in the Data Objects grid. The grid also shows the Row Hit Rate, Value Hit Rate, and Selectivity for each join.
The statistically strongest join is selected as Primary. It is indicated by a green button. The other relationships in the list are alternative possibilities that were discovered but rejected as too weak, based on statistical analysis.

**Data Objects Grid Tools**

The following standard tools are available for viewing and manipulating the data in the Data Objects grid. Also see the [Grid Panes](#) description for common actions you can perform in a grid.

- Add a data object
- Delete a data object
- Refresh Statistics after adding or modifying a data object
- Column Chooser
- Data Objects Data Preview – Show Hits, Show Misses

**Data Object Join Statistics**

Each expression in the alternatives list is accompanied by statistics for the parent and child tables.

- Row Hit Rate
- Row Hit Count
- Value Hit Count
- Selectivity

These statistics are an indication of how strong or weak the relationship is. Use this information to evaluate and approve the data object before proceeding.

**Data Objects Data Preview – Show Hits, Show Misses:**
About this task

When you select a data object in the Data Objects Diagram and click Show Hits or Show Misses, the Data Preview screen displays the rows in the parent and child tables that match or do not match the selected expression.

You can also Select Preview Criteria.
Chapter 3. Discovering data associations

The goal of source data discovery (SDD) projects is to discover overlaps or create a unified schema.

Discover Overlaps

The Overlaps tab only appears in Source Data Discovery Projects.

Overview

Navigation Map

Navigating in the Overlaps step is performed by clicking active links and icons to Drill Down and Up. Not all columns in a grid have active links; most data is read-only.

A diagram showing only the rows in each view that have active links, and the destination of each link, is provided below.

Drill Down and Up

About this task

The Overlaps screen displays data in two tabs: Summary Views and Overlaps Views. Each tab displays data at the Data Set level by default, then drills down to the Table level, and then to the Column level. Some links display the Column level directly from the Data Set level.

The Navigation Map shows how the views link to each other.
Drill down to a lower level in a tab by clicking on an active link in the grid, and return to a higher level in the tab by clicking the Back icon.

In the above example, clicking 3 in the Data Set 2.HQ_STORES row and the Data Set 1.HQ_ADDRESSES column displays those three overlapping rows in the Column Overlaps view, shown below.

To return to the Table Overlaps view again, click Back. Clicking Back once more displays the highest view in the Overlaps tab, Data Set Overlaps.

There are several exceptions to using Back to display the previous view.
- For links that open the Column Analysis window, click back on the Overlaps tab to continue.
- For links that open the Value Frequencies popup, close the popup to continue.
- In the Data Set Overlaps and Table Overlaps views, some links open the Column Summary view. Click on the appropriate view (Data Set Overlaps and Table Overlaps) to go back.

Obtaining Cleaner Value Overlap Results
There are several ways to reduce invalid Overlap Value results, reduce the amount of “noise” in the results, and reduce the amount of time needed to run the Overlaps task.
- Start with the cleanest data possible, to avoid false matches or missed matches.
- In each sample set, Set Columns to Null for columns containing data that you know is not relevant to the project. This eliminates processing on those columns and speeds up discovery.
- Mark columns as Critical Data Elements (CDEs) during Column Analysis, and then use Show All Items in the Overlaps screens to filter non-CDE columns from the display.
As you work through the Overlaps results, continue to set columns as Critical Data Elements (CDEs) and filter the displays to focus on relevant columns only.

In the Value Overlap Details verify that the overlap statistics shown in the Column Summary are valid matches. Make changes as necessary, including deleting results that are invalid or that are poor matches. Refresh Statistics after deleting results.

Overlap Examples
Overlap examples are shown below.

Overlap Example 1:

This example is for a Data Set 1 value overlap with Data Set 2.

The selected data set in this Column Summary view is Data Set 1. The first row contains data for HQ_ADDRESSES.EMPLOYEE_ID in Data Set 1. This column has been marked as a critical data element, meaning it was manually identified as being important in this project. If desired, you can change Show All Items to Show Items with CDEs, to filter out all non-CDE columns from this list.

The first row of data in Value Overlap with Data Set 2 indicates that when the values in Data Set 1’s HQ_ADDRESSES.EMPLOYEE_ID are compared to a particular column in Data Set 2, there is a 93% value overlap. Note that the results of comparing values in this direction are shown under the Data Set 1 header.

These two columns are also compared in the reverse direction. When the values in that Data Set 2 column are compared to Data Set 1’s HQ_ADDRESSES.EMPLOYEE_ID, there is a 90% overlap. The results of comparing values in this direction are shown under the Data Set 2 header.

The same Data Set 2 column is used in both statistics, but its name is not shown in this view. To identify the Data Set 2 column, click either 93% or 90% to display the Value Overlap Details. The following example shows the Value Overlap Details popup when 93% is clicked.

The Value Overlap Details popup shows that the Data Set 1 column HQ_ADDRESSES.EMPLOYEE_ID overlaps with only one column in Data Set 2,
which is HQ_SALARIES1.EMPLOYEE_ID. The Row Hit Rate, Unbound Rows Passing Filter, and Selectivity for the two columns are also shown, which can help in determining if this is a valid overlap relationship. The statistics are high, which indicates this may be a valid overlap.

To view the actual data in the two columns and the overlapping rows, or to view the rows that do not overlap, click Show Hits, Show Misses.

The Data Preview Show Hits window appears. Based on the high number of overlapping columns, and the fact that both tables contain data about employees, this value overlap can be considered a valid match.

Overlap Example 2:

This example is for a Data Set 1 value overlap with Data Set 3.

This Column Summary view compares the Data Set 1 column with all other data sets in the project. This sample project contains three data sets, so there is an entry
for Value Overlap with Data Set 3.

The first row of data in Value Overlap with Data Set 3 shows that the selected Data Set 1 column, HQ_ADDRESSES.EMPLOYEE_ID, has a 100% overlap with a particular column in Data Set 3. The result of comparing columns in this direction is shown under the Data Set 1 header.

The same columns are also compared in the opposite direction. When that Data Set 3 column is compared to Data Set 1’s HQ_ADDRESSES.EMPLOYEE_ID, there is also a 100% overlap, shown under the Data Set 3 header. Click on either 100% link to display the Value Overlap Details.

The same Data Set 3 column is used in both statistics, but its name is not shown in this view. To identify the Data Set 3 column, and to see if there were any other overlaps discovered, click either value overlap link to display the Value Overlap Details. The following screen shows the Value Overlap Details popup when the Data Set 1 100% link is clicked.

The Value Overlap Details popup shows that the Data Set 1 column HQ_ADDRESSES.EMPLOYEE_ID overlaps with two columns in Data Set 2: HQ_EMP1.EMPLOYEE_ID and HQ_ADDRESSES1.EMPLOYEE_ID. The Row Hit Rate, Unbound Rows Passing Filter, and Selectivity for the two columns are shown. The overlap with HQ_ADDRESSES1.EMPLOYEE_ID yields a higher row hit rate and value hit rate than the overlap with HQ_EMP1.EMPLOYEE_ID.

To view the actual data in the two columns and the overlapping rows, or to view the rows that do not overlap, click Show Hits, Show Misses. By comparing the hits and misses between both overlapping results, you can determine if both overlap matches are valid.

Overlap Example 3:

This example demonstrates how to investigate results.

Assume a sample project containing four data sets: HR, Payroll, Benefits, and Skills. The steps below show you how to identify potential CDEs in the data, identify significant inconsistencies in the data, and finally, identify the most trustworthy source.

Step 1: Identify Potential CDEs:
Performing this action allows you to focusing on the CDEs, removing unnecessary data from the view.

- In the Overlaps tab, the Overlap grid displays the sources that are most prominently overlapping with other sources. Investigate these further by drilling down into the data, and if they look promising, mark them as CDEs.

- Use the Total Columns information for each source to review the statistics for all sources. Columns that overlap with many sources at a high hit rate are potential CDEs. Investigate them further and mark them as CDEs if they appear important.

**Step 2: Identify Significant Inconsistencies:**

Assume the column EMP_ID is marked as a CDE. Using the Column Summary data, you see that HR.EMP_ID has more IDs than Payroll.EMP_ID. Is this an expected result? Looking at the Address fields in these tables reveals many inconsistencies. Is there an explanation for this? Some department IDs in Payroll are missing from the HR table. Does this mean that money is being paid from wrong departments? All these could lead to an ROI driver for MDM.

**Step 3: Identify The Most Trustworthy Source:**

We need to determine which source contains more EMP_IDs data with fewer discrepancies. The HR data for all other columns is 100%, which is very good. Can HR provide all the other attributes needed? To determine this, perform the following actions.

- Check all the exclusive attributes in HR and Payroll. For example, Payroll has 6 exclusive columns that do not appear in HR. Are any of these columns required?
- Review all elements. If overlaps exist with other sources, are the HR versions the best? For example, a column named HSA_ACCT may not compare well to a similar column in the other sources. Even though HR contains all of the EMP_ID values, this may indicate that HR is not trustworthy for HSA information. By drilling down into the HSA_ACCT results, you see that it matches perfectly with Payroll but not with HR. This may mean that Payroll is more up-to-date on HSA information than HR is.

**Discover Overlaps**

You do not need to explicitly run Column Analysis, PF Keys, or Data Objects discovery before running Overlaps discovery. If you run Overlaps discovery without previously explicitly running Column Analysis, Discovery will automatically discover Column Analysis and Overlaps data only.

**Procedure**

2. If desired, click **Sub Steps** and select **Column Classification: Value Matching**.

3. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   **Note:** Modifying task options is an advanced activity. Options are described in [Reference of Task Options](#). Modifying options and creating new option sets is described in [Option Sets](#).

4. Click **Run** to execute Overlaps discovery.
   Discovery is performed on all tables and text files in the data sets. If the procedure does not complete correctly, see [Troubleshooting Task Execution](#).

   **Note:** Although the Overlaps window is displayed immediately, it may take some time for the actual data processing to complete. Check the [Activity Viewer](#) for the status of the task.

### Re-run Overlaps

#### About this task

The **Re-run Step** button allows you to perform or repeat Overlaps discovery from within the Overlaps window. Any changes you have made, such as marking columns as **Critical Data Elements (CDEs)** will be kept.

**Note:** You can run Overlaps discovery without previously running Column Analysis, PK Keys, or Data Objects. Doing so will discover Column Analysis and
Overlaps data only.

Use **Re-run Step** in the Overlaps window if:

- You change any Overlaps task options and want to see how the changes affect discovery.
- You modified any Column Analysis data, such as data types or length.
- The data sets have changed in any way, such as reloading tables or changing sample sets.

**Note:** If you modify a data set and re-run Overlaps without also re-running PF Keys and Data Objects, the Overlaps and Column Analysis results will be updated, but the PF Keys and Data Objects results will not be modified and may be incorrect.

To re-run Overlaps processing:

**Procedure**

1. Click **Re-Run Step** in the Overlaps window. The Processing Options Dialog appears.
2. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.
3. Click Run. Discovery discovers Overlaps for all data sources (database tables and text files) in the data sets.

Note: Although the Overlaps window is displayed immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the task status.

Delete Invalid Overlap Results

About this task

Deleting invalid overlap results can reduce clutter in the Value Overlap Details popup. To delete an overlap match, click in the row and then click Delete.

If you delete the overlap result that was displayed in the Column Summary, Refresh Statistics in Column Summary to update the display with the next-highest overlap result.

Review Overlaps

The main task in the Overlaps tab is to review the discovered overlaps. This includes viewing the column data to verify that the overlaps are useful and valid, and deleting incorrect overlaps. You can also define new overlaps and refresh the statistics to view the results.

You can sometimes identify poor overlaps based on the column names involved. For example, Discovery might identify a strong data match between the SHIPPER_ID and EMPLOYEE_ID columns of two tables. Columns of this type often consist of sequential integers, so numerically, there may be a very strong overlap between such columns, but logically the overlap is meaningless.

As you review the overlaps, delete any obvious mismatches in the Value Overlap Details. If there is any doubt about the data in a particular overlap, use Column Summary and Column Overlaps to display the actual data.

Understanding the Value Overlap Results

The Column Summary view displays some statistics taken directly from the Column Analysis tab. It also displays the percentage of overlap that each selected column has with particular columns in one or more of the other data sets in the project. The number of columns shown and the number of data sets they are compared to varies depending on where you displayed Column Summary from, but the statistics are calculated in the same manner.
For example, the screen above displays the **Column Summary** for Data Set 1 when displayed from Data Set Summary. The first group of statistics is taken from the Column Analysis page for Data Set 1. There are also two groups of statistics showing value overlaps with the other two data sets in the project. These are the Value Overlap columns.

There are several things to understand about the Value Overlap columns:

- Displaying the Highest Value Overlaps
- Value Overlap Statistics
- Delete Invalid Overlap Results

**Displaying the Highest Value Overlaps**

The **Column Summary** displays the results for only one data set column pair per row. The selected column may have value overlaps with several (or many) columns in the other data sets, but only the column pair with the highest overlap values, or the values that Discovery considers the “best”, is displayed in the **Column Summary**.

**About this task**

To see if any other column pairs were discovered, click a Value Overlap statistic. The Value Overlap Details appears showing all columns in the other data set that have overlapping results with the selected column. In many cases, it will be obvious if the overlap is valid or not, based on the table and column name. In other cases, you may need to preview the data to determine whether the overlap is valid or not.

If there are many matches, deleting invalid overlap results can make it easier to focus on the valid results.

**Value Overlap Statistics**

The actual overlap values are shown for each column in the selected data set and the most closely overlapping column Discovery could find in the other data set(s), as described in **Displaying the Highest Value Overlaps**.

**The Overlaps Window**

The Overlaps window displays all discovered overlaps between the data sets in the project.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Column</th>
<th>Value Overlap with Data Set 2</th>
<th>Value Overlap with Data Set 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set 1</td>
<td>STAND NO</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>Data Set 1</td>
<td>STAND NO_</td>
<td>32</td>
<td>32</td>
</tr>
</tbody>
</table>
You cannot add, edit, or delete any values or rows in the Overlaps views themselves; they provide an interactive, read-only display of the column overlaps within the project. However, you can mark and unmark columns as Critical Data Elements (CDEs) in the Column Overlaps view. You can also delete overlaps in the Value Overlap Details, which can change the values displayed in the Overlaps views.

Common Window Elements describes the menus, tabs, and buttons (like Error List and Activity Viewer) that are common to all windows.

The parts of the screen are described below.

- The **Summary Views** and **Overlaps Views** display discovered statistics about the data sets, tables, and columns in the project. The name of the currently-displayed data set or table is shown above the tab. When the screen displays Table Overlaps or Column Overlaps, the names of both data sets or tables are shown.
- The **Top** icon returns both views (Summary and Overlaps) to the top level.
- The **Back** icon displays the next-highest view in the tab.
- **Show All Items** filters the grid to display all results, statistics only for columns marked as Critical Data Elements (CDEs) or statistics only for columns not marked as CDEs.
Show Overlaps/Show Non-overlaps allows you to view either statistics about the overlapping columns in the project, or statistics about the non-overlapping (exclusive) columns in the project. This option is valid only in Overlaps Views.

Summary Views

The Summary views display general statistical information about the tables, columns, and rows in the project.

There are three Summary views:
- Data Set Summary
- Table Summary
- Column Summary

Data Set Summary

The Data Set Summary view displays the total number of tables, columns, Critical Data Elements (CDEs) and exclusive (non-overlapping) columns in each data set in the project.

The screen shot below shows the Data Set Summary view for a project containing two data sets.

This view is the only one that includes a chart representation of the data. Each component of the chart represents a column in the grid. If there are many data sets or tables, you may need to Resize the Discovery Studio window to view the entire chart.
Data Set Summary Grid Tools:

The following standard tools are available for viewing and manipulating the data in this grid. Also see the Grid Panes description for common actions you can perform in a grid.

- **Search**
- **Column Chooser**
- **Export Grid Content**
- **Refresh Statistics** updates the statistics in the grid after deleting any results in The Overlaps Window.

The columns in the grid are:

- **Data Set**: Lists each data set in the project.
- **Total Tables**: Total number of tables in each data set. Click a value to display the Table Summary view with column, CDE and row data for each table in the selected data set.
- **Total Columns**: Total number of columns in each data set. Click a value to display the Column Summary view with statistical information for each column in the data set, including CDEs and overlaps with other data sets.
- **Total CDE’s**: Total number of columns flagged as Critical Data Elements (CDEs) in the data set. Click a value to display the Column Summary view with statistical information for the CDE columns in the data set.
- **Total Overlapping Columns**: Total number of columns in the data set that overlap with columns in other data sets in the project. Click a value to display the overlapping columns in the Column Summary view.
- **Total Exclusive Columns**: Total number of columns in the data set that are unique (non-overlapping). Click a value to display the Column Summary view showing the total number of columns in the data set that do not overlap with any other columns in the project.

**Table Summary**

About this task

The Table Summary view displays the number of columns, rows, and Critical Data Elements (CDEs) in each table in the selected data set.

The name of the data set is shown above the tabs. The screen below shows data about the data set Source1.

```
<table>
<thead>
<tr>
<th>Data Set</th>
<th>Table</th>
<th>Columns</th>
<th>CDE's</th>
<th>Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source1</td>
<td>HQ_ADDRESSES</td>
<td>7</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>Source1</td>
<td>HQ_DEPARTMENT</td>
<td>4</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>Source1</td>
<td>HQ_EMP</td>
<td>11</td>
<td>0</td>
<td>31</td>
</tr>
</tbody>
</table>
```
To display this information for a different data set, click Back to display the Data Set Summary and then click a different data set’s Total Tables value.

Table Summary Grid Tools:

The following standard tools are available for viewing and manipulating the data in this grid. Also see the Grid Panes description for common actions you can perform in a grid.

- Show All Items
- Search
- Column Chooser
- Export Grid Content
- Refresh Statistics updates the statistics in the grid after deleting any results in The Overlaps Window

The columns in the grid are:

- **Data Set**: Lists each data set in the project.
- **Table**: Name of the table.
- **Columns**: Number of columns in each table. Click a table’s Columns value to display the Column Summary view with statistical information for all columns in the data set, including CDEs and overlaps with other data sets.
- **CDE’s**: Number of columns flagged as Critical Data Elements (CDEs) in the table. Click a table’s Total CDEs value to display the Column Summary view with statistical information for all CDE columns in the table.
- **Rows**: Number of rows in the table.

Column Summary

The Column Summary view displays profile statistics and Critical Data Elements (CDEs) status for the columns in the selected table, along with the overlap percentage with the other data sets in the project.

This view can be displayed from the following tabs (see Navigation Map). Depending on which tab you start from, the number of columns displayed and the overlap statistics are slightly different.

- Displayed from Data Set Summary
- Displayed from Table Summary
- Displayed from Data Set Overlaps
- Displayed from Table Overlaps

Displayed from Data Set Summary:

When this view is displayed from the Data Set Summary view, it lists all of the columns in the selected data set. The name of the selected data set is shown above the tabs. Some calculated statistics about the columns are shown, taken directly from the Column Analysis screen.

About this task

The last columns in the grid display the percentage of value overlap, if any, for each column in this data set, compared to each other data set in the project, in each direction. For a full description of these columns, see Understanding the Value.
To display statistics and overlap information for each column in a different data set, click **Top** to display the [Data Set Summary] then click a different data set's **Total Columns** value.

**Displayed from Table Summary:**

About this task

The statistics shown when **Column Summary** is displayed from the **Table Summary** view are a subset of the statistics described in **Displayed from Data Set Summary**. Instead of all of the columns in the selected data set, the view contains only the columns in the selected table.

**Displayed from Data Set Overlaps:**

When this grid is displayed from the **Data Set Overlaps** view, it lists *only the overlapping columns* between two specific data sets. The names of the two data sets are shown above the tabs. Some calculated statistics about the columns are shown, taken directly from the Column Analysis screen.

About this task

The last columns in the grid show the percentage of value overlap for each displayed column, when compared with the other data set in each direction.

The sample screen below shows the overlapping columns between Data Set 2 and Data Set 1. The last columns show the percentage of overlap that *each overlapping Data Set 2 column has with Data Set 1, in each direction*. For a full description of these columns, see [Understanding the Value Overlap Results]
To display column overlap information for a different pair of data sets, click the **Overlaps Views** tab to re-display the **Data Set Overlaps** view, then click a different **Columns** value.

**Displayed from Table Overlaps:**

**About this task**

The statistics shown when **Column Summary** is displayed from the **Table Overlaps** view are a subset of the statistics described in the **Displayed from Data Set Overlaps**. Instead of listing all of the overlapping columns in the two selected data sets, the view contains only the columns in the two selected tables which contain overlapping values.

**Column Summary Grid Tools:**

The following standard tools are available for viewing and manipulating the data in this grid. Also see the **Grid Panes** description for common actions you can perform in a grid.

- Show All Items
- Search
- Column Chooser
- Export Grid Content
- **Refresh Statistics** updates the statistics in the grid after deleting any results in The Overlaps Window

Many columns in the grid contain statistical data from the Column Analysis screen.

The columns in the grid are:

- **Data Set:** The selected table’s data set.
- **Table:** The selected table.
- **Column**
- **Column Number**
- **CDE**
- **Classification (Group Names)**
- **Cardinality**
- **Rows**
- **Selectivity**
- **Non Nulls %**
• **Value Overlap with [Data_set_name]**: The percentage of overlap, if any, that the column in the selected data set has with each remaining data set in the project, in each direction. For a full description of these columns, see [Understanding the Value Overlap Results](#).

**Value Overlap Details**

**About this task**

The **Value Overlap Details** list is a popup listing all discovered overlap relationships between the selected column and columns in another data set. Display the list by clicking an overlap percentage in the **Column Summary** view or **Column Overlaps** view.

You can preview the actual data in the two columns by selecting the row in the popup and clicking **Show Hits, Show Misses**. This can help you determine if the overlap relationship is valid.

You can delete relationships in this list. If you do, [Refresh Statistics](#) after you close this list. You can also add relationships.

The **Value Overlap Details** popup uses a filter to display only the rows associated with the value you clicked on. Any rows you add in this screen need to pass the filter in order to be displayed in the screen. You can click on the filter and edit it to display other results.

**Note**: After making the desired changes in this popup (adding a row, deleting a row, or refreshing statistics), close the popup and then *immediately save the project*. If you do not, the changes will be lost.

The standard tools for **Grid Panes** are available for viewing and manipulating the data in the **Value Overlap Details** popup. In addition, there are two more tools:

- Change the sort order for columns displaying both percentage and fraction statistics using **Sort by Rate** or **Sort by Count**. For more information, see [Show All](#).
- Copy one column’s classification to the other column using **Classify Left Columns** or **Classify Right Columns**.
(If the current data set is the first data set in the pair, the tool is set to Classify Right Columns. If the current data set is the second data set in the pair, the tool is set to Classify Left Columns.)

The list contains the following information about the columns:

- **Type**
- **Left Data Set, Table, Column, Classification (Group Names)**: the data set, table name, column name, and classification of one table/column in the pair.
- **Right Data Set, Table, Column, Classification (Group Names)**: the data set, table name, column name, and classification of the other table/column in the pair.
- **Data Rule**: the data rule, if any, used by this relationship
- **Row Hit Rate**
- **Value Hit Rate**
- **Selectivity**
- **Data Types, Mixed**: Indicates whether the column contains both numeric and nonnumeric data types. When a column has mixed data types, preview the data to determine whether the results are accurate.

**Overlaps Views**

The Overlaps views display data about the overlapping columns in the data sets and tables.

There are three Overlaps Views:

- **Data Set Overlaps**
- **Table Overlaps**
- **Column Overlaps**

**Data Set Overlaps**

The Data Set Overlaps view displays the total number of tables and columns, and the number of overlapping tables and columns, in each data set in the project.

The number of overlapping columns can be different for the same pair of data sets, depending on which data set is compared to the other. This grid uses the data sets in the Data Set column as the basis for comparison with the other data sets in the project.

For example, the Data Set Overlaps view below shows that Data Set 1 contains 29 columns, and 11 of them overlap with Data Set 2. Data Set 2 has 17 columns, and 9
of them overlap with Data Set 1.

Data Set Overlaps Grid Tools:

The following standard tools are available for viewing and manipulating the data in this grid. Also see the Grid Panes description for common actions you can perform in a grid.

- Show All Items
- Show Overlaps/Show Non-overlaps
- Search
- Column Chooser
- Export Grid Content
- Refresh Statistics updates the statistics in the grid after deleting any results in The Overlaps Window

The columns in the grid are:

- **Data Set**
- **Total Tables**: Total number of tables in each data set.
- **Total Columns**: Total number of columns in each data set.
- **[Data_set_name] Value Overlap**: The number of overlapping tables and columns, if any, that the selected data set has with each other data set in the project.
  - **Tables**: The number of tables in this data set that contain columns overlapped by the other data set. Click the value to display the Table Overlaps view showing each table that contains overlapping columns.
  - **Columns**: Total number of columns in this data set overlapped by columns in the other data set. Click a value to display the Column Summary view with statistical information for each overlapping column in the data sets.

Table Overlaps

This view shows, for the tables in one data set, the number of columns that overlap when compared to the columns in a second data set.

About this task

The names of the two data sets are shown above the tabs. The screen below shows the columns that overlap when the data set Source1 is compared to the data set Source2.
To display this information for a different pair of data sets, click Back to display the Data Set Overlaps view and then click a different data set's [Data_set_name] Tables value.

Table Overlaps Grid Tools:

The following standard tools are available for viewing and manipulating the data in this grid. Also see the Grid Panes description for common actions you can perform in a grid.

- Show All Items
- Show Overlaps/Show Non-overlaps
- Search
- Column Chooser
- Export Grid Content
- Refresh Statistics updates the statistics in the grid after deleting any results in the Overlaps Window

The columns in the grid are:

- **Table**: Each table in the first data set to be compared.
- **Columns**: Number of columns in each table in the first data set. Click a table's Total Columns value to display the Column Summary view with statistical information for the overlapping columns in the table.
- **[Data_set.Table_name]**: The grid contains one column for each table in the second data set. Click a value in this column to display the Column Overlaps view showing the overlapping columns between these two tables.

Column Overlaps

The Column Overlaps view displays statistics and overlap percentages for the overlapping columns between two tables in different data sets.

About this task

The names of the two data sets and tables being compared are shown above the tabs. The following screen shows the overlapping columns between HQSTORES from Data Set 2, and HQ_ADDRESSES from Data Set 1.
To display overlapping column information for two other tables in different data sets, click Back to display the Table Overlaps view, then click a value for a different pair of tables.

**Column Overlaps Grid Tools:**

The following standard tools are available for viewing and manipulating the data. Also see the Grid Panes description for common actions you can perform.

- Show All Items
- Show Overlaps/Show Non-overlaps
- Search
- Column Chooser
- Export Grid Content
- Refresh Statistics updates the statistics in the grid after deleting any results in The Overlaps Window

Many columns in the grid contain statistical data from the Column Analysis screen. The columns in the grid are:

- Data Set
- Table: The first table's name.
- Column: The name of each column in the first table that overlaps with columns in the second table.
- Column Number
- CDE: Allows you to flag columns as Critical Data Elements (CDEs)
- Cardinality
- Rows
- Selectivity
- Non Nulls %
- [Table_name]: The percentage of overlap that each column in the first table has with the second table. Click a value to display the Value Overlap Details popup.

**Define Target Table Schemas**

A unified schema is a virtual table with a schema that unifies all tables mapped to it.

For an introduction to unified schemas and a description of the steps involved in creating a unified schema, see Create a Target Table for a Unified Schema.
Overview

A project can contain multiple target tables, but you can only edit one target table at a time. Each target table is independent: changes made to one target table do not affect data in any other target table. You cannot combine one target table with another, or include a column from one target table in another target table.

After defining a target table schema, you define mappings from the data set’s logical tables, including, if necessary, how to extract data from each data set to the target table.

After the target table is created, Discovery offers powerful tools to analyze your consolidated data and prototype the Match and Merge.

A unified schema is a virtual table with a schema that unifies all tables mapped to it. This section describes the unified schema builder.

Creating Target Table Schemas

About this task

To define a target table is to identify columns of the target table. If you already have a schema you want to use, you know the columns. If not, one way to quickly build a target table schema is to use Critical Data Elements (CDEs).

In the Target Table Schema Window filter the column grid to only display CDEs and then drag the source tables or columns into the definition pane. It is not necessary for a particular target table to include all of the marked CDEs, and you can add non-CDE columns to the target table if appropriate.

There are several ways to create target table schemas.

- Define a Target Table Schema
- Import a CWM File as a Target Table Schema
- Copy a Target Table Schema

Define a Target Table Schema

You can define a target table schema.

Procedure

Click the Unified Schema tab.

- If this is the first target table in the project, no sub-tabs appear.
- If at least one target table exists, the window opens to the last used tab. Click the Target Table Schema tab.

Create an Empty Target Table Schema:

Procedure

1. Click the Add icon.
A new, empty target table is created, and the **Target Table Schema Grid** appears. (If this is the first target table in the project, all of the Unified Schema tabs now appear.) All of the tables and columns from all data sets in the current project are listed in the **Data Sets Column Grid**.

2. Rename the new target table, if desired, by clicking in the **Name** field and typing a new name.

**Add Columns to the Target Table Schema:**

**Procedure**

1. Add target columns to the target table. Create one target column for each data element. For example, one target column may be populated with data from a single table column named **HIRE_DATE**, while another target column may be an aggregate of three table columns named **SOCIAL**, **S0SECNO**, and **ID**.

   Target columns are added to a target table in several ways:

   - Click the + icon above the **Target Table Schema Grid** to add a blank target column, as shown below:
Click on a column from the **Data Sets Column Grid** and then drag it into the Target Table grid, as shown below. (Shift-click to select adjacent rows, or control-click to select nonadjacent rows for dragging.) One target column is created for each dragged table column, with the name and data type of the original table column.

If a target column of the same name already exists, a sequential number is added to the newly dragged column name to make it unique.

Click on a table name and drag it into the Target Table Grid, as shown below. One target column is created for each displayed column in the table, with the name and definitions of the original columns.

If any dragged columns have the same name as an existing table column, a sequential number is added to the newly dragged column name to make it unique.
Filter the Data Sets Column Grid so that only the desired columns are displayed in a table or in the entire list, and then drag each table into the Target Table grid. In the example below, the data set is filtered by Checked CDE. You can also shift-click the filtered list to select adjacent rows, or control-click to select nonadjacent rows for dragging. One target column is created for each table column, with the name and data type of the original table column.

If any dragged columns have the same name as an existing table column, a sequential number is added to the newly dragged column name to make it unique.
2. Give the target column a descriptive name. Each target column in this target table must have a unique name.

3. If needed, set or change the target column's Data Type, Length and other characteristics.

4. If desired, rearrange the order of the target columns as follows:
   a. Click the # column header to sort it in ascending order.

   b. Select the desired target column's sort number in the # column and drag it just below its new location in the list.
c. Reorder any other target columns as needed.

5. To view transformations currently mapped to target column, click the + icon by the target column name. The Transformations field displays the SQL expressions used to extract data from the table columns. To view the statistics for each transformation, use the Column Chooser to display the Statistics band.

When a target column was created by dragging a table column into the Target Table grid, the table and column name are automatically used as the transformation. All other data sets have NULL expressions by default, meaning no mapping transformations are provided from other data sets to the target column.

In the Source Mapping Definition Grid, you can change existing expressions. You can also map additional table columns from other data sets with this target column. Changes made in that screen are immediately updated here.
6. After the target table contains all required target columns, and each target column's characteristics have been set, continue with Define Source Mappings.

**Import a CWM File as a Target Table Schema**

A CWM file can be imported as target table schemas. This is useful for importing a known model into a project, rather than creating it from scratch.

**About this task**

When an imported CWM file contains primary-foreign keys, they are imported along with the tables and are visible in The Unified Schema PF Keys Window. However, the PF Key statistics cannot be viewed until the target table is mapped and populated.

**Procedure**

1. Create the CWM file that will be imported. The file can contain multiple target tables.

2. In the Target Table Schema tab, click the Import CWM icon.
3. In the file browser, navigate to the location of the exported CWM file, select it, and click **Open**.

Discovery creates a target table for each target table in the CWM file. The rows are not mapped to any source table columns. You can now perform source mapping.
Copy a Target Table Schema

About this task

You can duplicate a target table. This makes an exact copy of the target table, including all mappings.

To copy a target table, select the table in the list and then select Tools>Unified Schema Table Copy.

A copy of the table is created and named Copy_Of_<tablename> by default. You can change this name.

Export Target Table Schemas as CWM

Target tables can be exported in CWM format. All target tables and Unified Schema PF Keys in the project are included in the export.
For instructions, see Export CWM XML.

**Target Table Schema Window**

When you click the Unified Schema tab, the Target Table List appears. No sub-tabs appear until you Define a Target Table Schema.

After at least one target table is created, the Unified Schema sub-tabs are available.

Common Window Elements describes the menus, tabs, and buttons (like Activity Viewer and Error List) that are common to all windows.
The Target Table Schema tab contains the following areas.

- Target Table Schema List
- Target Table Schema Grid
- Data Sets Column Grid

**Target Table Schema List**

The Target Table List displays all of the target tables in the project, with statistics about each target table and the columns used by each target table. This list is empty until you **Define a Target Table Schema**.
The screenshot below shows a sample Target Table list. (The list border has been dragged to display all columns.) This list contains two target tables, with the first target table expanded to display the contents. In this example, the target table EMPLOYEES uses columns from all three data sets in the project.

In the target table EMPLOYEES, 10 out of 13 target columns have non-null mappings (transformations) from the HQ data set. One non-null column (out of 13) is from the West data set, and two non-null columns are from the Northwind data set.
Target Table Schema List Tools

The following standard tools are available for viewing and manipulating the data in the Target Table list. Also see the Grid Panes description for common actions you can perform in a grid.

- Auto Filter Row
- Column Chooser
- Export Grid Content
- Define a Target Table Schema (+ icon)
- Delete a target table

Target Table Schema List Columns

The Target Table list displays data about each target table in the current project. This data is updated immediately whenever changes are made.

- Name (Unified Schemas) (editable field)
- Mapped Data Sets (Unified Schema)
- Mapped Row Count (Unified Schema)
- Approved Columns
- Notes

When a target table is expanded, the following information is displayed about the target table. This information is updated immediately whenever changes are made.

- Name
- Non-null Columns in each data set
- Notes

Target Table Schema Grid

The Target Table grid lists each column that has been added to the selected target table. The Statistics band is hidden by default; use the Column Chooser to display the statistics for each column discovered during Column Analysis.
The values in the Target Table band can be modified. Because the target table is staged as a new, separate table, changing these values does not modify the values in the Column Analysis Window or in any source tables. However, if you do change these values, you should re-run all Unified Schema Builder steps.

**Target Table Schema Grid Tools**

The following standard tools are available for viewing and manipulating the data in the Target Table grid. Also see the Grid Panes description for common actions you can perform in a grid.

- Advanced Search
Search
Auto Filter Row
Column Chooser
Export Grid Content
Add a column from a table to the selected target table
Delete a column from the selected target table

Target Table Schema Grid Information

The Target Table grid displays information about the columns in the target table. Additional statistics are available from the Column Chooser. The information is divided into the Target Table Band and the Target Table Schema Statistics Band.

The columns in the Target Table grid are described below.

Target Table Band:

Most of these values can be edited. This does not affect the source tables.
- Approved
- #
- Column Name
- Data Type
- Length
- Precision
- Scale
- Notes

Target Table Schema Statistics Band:

(Available from the Column Chooser) After the Unified Column Analysis step is run, this band displays statistics calculated for each target column based on the Column Analysis data for each mapped source column.
- Mapped Data Sets (Unified Schema)
- Cardinality
- Selectivity
- Min
- Max
- Null Count (available from Column Chooser)
- Non Null Rows (available from Column Chooser)
- Non-null Selectivity (available from Column Chooser)

When a column in the Target Table grid is expanded, the following information is displayed about the column:

Mapping Band:
- Data Sets (Unified Schema)
- Transformation (Unified Schemas) as defined in the Source Mapping tab

Mapping Statistics Band:
After the Unified Column Analysis step is run, the Mapping Statistics band displays statistics calculated for each source column mapped to the target column, based on the Column Analysis data for the source column. This band is not shown by default; display it using the Column Chooser.

- **Cardinality**
- **Selectivity**
- **Min**
- **Max**
- **Null Count** (available from Column Chooser)
- **Non Null Rows** (available from Column Chooser)
- **Non-null Selectivity** (available from Column Chooser)

**Data Sets Column Grid**

**About this task**

The Data Sets Column Grid is displayed in the **Target Table Schema Window**, **Source Mapping Window**, and **Match and Merge Analysis Window**.
This grid contains a read-only summary of every source table in each data set in the project, along with the Column Analysis statistics for each source column. When a source table in the grid is filtered and the table is dragged into the center grid, only the visible columns in the table are available.

- In the **Target Table Schema Window** all tables in all data sources are shown. Dragging a table column from here into the **Target Table Schema Grid** maps the table column to a new target table column. Dragging a source table into the **Target Table Schema Grid** maps all visible source table columns to new target table columns.

- In the **Source Mapping Window** the tables and columns in the selected data set are shown. Dragging a source table or column from here into the Expression Editor adds that source table or column to the selected expression.

- In the **Match and Merge Analysis Window** all tables in all data sources are shown. Dragging a source table or column from here into the Expression Editor adds that source table or column to the selected matching condition.

### Data Sets Column Grid Tools

The following standard tools are available for viewing the data in the Data Sets Column Grid. Also see the **Grid Panes** description for common actions you can perform in a grid.

- Advanced Search
- Search
- Auto Filter Row
- Column Chooser
- Export Grid Content

---

<table>
<thead>
<tr>
<th>CDE</th>
<th>Column</th>
<th>Classif...</th>
<th>Map...</th>
<th>Mapped To</th>
<th>Data Type</th>
<th>Card...</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set: HQ_ADDRESSES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMPLOYEE_ID</td>
<td>1 EMPLOYEE_ID</td>
<td>numberstrin...</td>
<td>30</td>
<td>10001</td>
<td>10031</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STREET_ADDRESS</td>
<td>1 STREET_ADD...</td>
<td>varchar(30)</td>
<td>32</td>
<td>100 FOPS ST.</td>
<td>ENDEHALL HOLLOW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CITY</td>
<td>1 CITY</td>
<td>varchar(30)</td>
<td>6</td>
<td>ATLANTA</td>
<td>TACOMA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COUNTRY</td>
<td>1 COUNTRY</td>
<td>varchar(30)</td>
<td>1</td>
<td>USA</td>
<td>USA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POSTAL_CODE</td>
<td>1 POSTAL_CODE</td>
<td>numberstrin...</td>
<td>10</td>
<td>95128</td>
<td>95403</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AREA_CODE</td>
<td>1 AREA_CODE</td>
<td>numberstrin...</td>
<td>7</td>
<td>206</td>
<td>710</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHONE_NUMBER</td>
<td>Sensitive</td>
<td>1 PHONE_NUMB...</td>
<td>varchar(10)</td>
<td>31</td>
<td>246-2319</td>
<td>929-2471</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Data Set: HQ_DEPARTMENTS | | | | | |
| DEPARTMENT_ID | 0 | varchar(5) | 12 | 100 | 3-100 |
| DEPARTMENT_NAME | 1 DEPARTMENT... | varchar(30) | 5 | ACCOUNTING | SECURITY |
| MANAGER_NAME | 0 | varchar(30) | 12 | DAMIEN FEE | WILLIE MCHUTCHISON |
| STORE_ID | 0 | varchar(10) | 6 | 51 | 36 |

| Data Set: HQ_EMP | | | | | |
| EMPLOYEE_ID | 0 | numberstrin... | 31 | 10001 | 10031 |
| TITLE_OF_CERTIFICATION | 0 | varchar(5) | 4 | OR | MS |
| NAME1 | id | varchar(40) | 27 | ALICIA | WILLIE |
| NAME2 | id | varchar(40) | 25 | BARTON | WILLIAM |
| SSN | Sensitive | 1 SSN | varchar(20) | 31 | 12-22-12 | 07-01-20 |
| DATE | Sensitive | 0 | datetime | 25 | 1997-09-19 00:00:00 | 1997-09-19 00:00:00 |
| DOB | Sensitive | 0 | datetime | 24 | 1997-01-11 00:00:00 | 1997-01-11 00:00:00 |
| STATUS | 0 | varchar(40) | 5 | CURRENT | RESIGNED |
| TERMINATION_DATE | 0 | datetime | 15 | 1996-02-09 00:00:00 | 1996-02-09 00:00:00 |
| RETURN_DATE | 0 | datetime | 7 | 1996-02-09 00:00:00 | 1996-02-09 00:00:00 |
Data Sets Column Grid Information
Source tables in the grid can be expanded and collapsed by clicking the + or - icon by the table name. The mapping information columns are updated immediately as you create the target table. The statistics are taken from the Column Analysis results for that source table.

About this task
- CDE
- Column
- Classification (Group Names)
- Mapped Count
- Mapped To
- Data Type
- Cardinality
- Min
- Max
- Length (available from Column Chooser)
- Precision (available from Column Chooser)
- Scale (available from Column Chooser)
- Mode (available from Column Chooser)
- Mode% (available from Column Chooser)
- Sparse (available from Column Chooser)
- Null Count (available from Column Chooser)

Perform Source Mapping

This chapter describes the source mapping step in the Unified Schema Builder tabs.

Overview
The Source Mapping tab allows you to build SQL queries that extract data from multiple data sets into the target table. These queries are built on logical tables, not on physical tables.

All standard SQL select statement restrictions apply. SQL queries created in the Source Mapping tab can use all of the standard elements used in SQL select statements:
- one select SQL expression (transformation) per target column
- join condition
- where clause
- group by list
- having clause (if group by is defined)

Suggest Transformations Tool Examples
The Suggest Transformations tool is a useful method of identifying potential transformations for source mapping. In order to use this tool, the Overlaps step must be completed. Several examples of using the tool are provided below.

Suggest Transformations Example 1:
For example, assume a target table `TrgTbl` that includes three different data sets (Data Set 1, Data Set 2, and Data Set 3) with one source table for each data set (Table1, Table2, and Table3).

Table1 from Data Set 1 is already mapped as shown:

The following diagram shows two kinds of relationships:
- mapping relationships between Data Set 1 and the target table
- overlap relationships between Data Set 1 and Data Set 2

The missing transformation is indicated by the red dashed line.
Chapter 3. Discovering data associations
Based on the existing transformations and the overlaps, the following transformations are suggested for Data Set 2:

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lee Harden</td>
<td></td>
</tr>
<tr>
<td>Bruce Lee</td>
<td></td>
</tr>
<tr>
<td>Mary Ann</td>
<td></td>
</tr>
<tr>
<td>Bruce Smith</td>
<td></td>
</tr>
</tbody>
</table>

The overlaps shown above, between Data Set 2 and Data Set 1, can be used because they are between columns in Data Set 2 and columns in Data Set 1 that already used in transformations. Only Data Set 1 is mapped so TrgTbl is the same as Data Set 1.

Note that transformations are suggested based on existing overlaps. Actual overlap numbers between a source and target can be greater or less than the suggested numbers.

- When using filters and or expressions, actual overlap numbers (between source tables) may be different than discovered during the Overlap step.
- Join conditions can affect the suggested overlap numbers.
- During the process of building the target table, overlap numbers can improve because more values are added to the target table. This often improves hit rates.

**Suggest Transformations Example 2:**

Now, assume that Data Set 1 and Data Set 2 are both mapped with Table 1 and Table 2. The goal is to map the FIRSTNAME column from Data Set 3 using the Suggest Transformation tool.

The following diagram shows two kinds of relationships:
- mapping relationships between Data Set 1 and Data Set 2
• overlap relationships between Data Set 3 and the other two data sets (Data Set 1 and Data Set 2) for FIRSTNAME

The missing transformation is indicated by the red dashed line.
<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>James</td>
<td>Smith</td>
</tr>
<tr>
<td>James</td>
<td>Jackson</td>
</tr>
<tr>
<td>Bruce</td>
<td>Smith</td>
</tr>
<tr>
<td>James</td>
<td>Taylor</td>
</tr>
<tr>
<td>Mary</td>
<td>Zach</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ann</td>
<td>Taylor</td>
</tr>
<tr>
<td>Lee</td>
<td>Xu</td>
</tr>
<tr>
<td>James</td>
<td>Smith</td>
</tr>
<tr>
<td>James</td>
<td>Wong</td>
</tr>
<tr>
<td>Ann</td>
<td>Osgood</td>
</tr>
<tr>
<td>Lee</td>
<td>Harden</td>
</tr>
<tr>
<td>Bruce</td>
<td>Lee</td>
</tr>
<tr>
<td>Mary</td>
<td>Ann</td>
</tr>
<tr>
<td>Bruce</td>
<td>Smith</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ann</td>
<td>Taylor</td>
</tr>
<tr>
<td>Lee</td>
<td>Xu</td>
</tr>
<tr>
<td>Edward</td>
<td>Smith</td>
</tr>
<tr>
<td>Roy</td>
<td>Wong</td>
</tr>
<tr>
<td>James</td>
<td>Taylor</td>
</tr>
</tbody>
</table>

Based on the existing transformations and the overlaps, the following transformations are suggested for Data Set 3:
The transformation TABLE3.FIRSTNAME is suggested for the target column FIRSTNAME.

In order to verify it, click on the link in the first row.

Note the following items of interest:

- The Show Details screen shows the overlaps (originally discovered during Overlap step) between the source column and the other columns used in the transformations. In this case, two overlaps are discovered for TABLE3.FIRSTNAME.
- The Row Hit Rate and the Value Hit Rate in the Suggested Transformations screen are the highest values from the Source Row Hit Rate and Source Value Hit Rate in the Show Details screen (circled in red).

The suggested transformation is based on the Overlap and Name criteria. The exact column name equality is used for Name suggestions.

A suggested transformation may qualify using the Overlap and Expression criteria (as described at the beginning of Suggest Transformations). This does not mean that the source and target column overlap, but the existing mapping can use expressions that are more complex than simple column references. In such a case, overlap numbers may be misleading. You can analyze the expression from the existing mapping, provided in the Expression column of the Suggested Transformation screen.

**Define Source Mappings**

**About this task**

When all of the necessary target columns are created (as described in Define a Target Table Schema), you are ready to define the source mappings for each data set that contributes to the target table. This can include:

- joining source tables from the data set that you will extract data from
- entering transformations (or select expressions) using columns from joined source tables
- filtering source data by defining where clauses
- grouping source data by defining group by lists and having clauses
By default, each target column has a transformation (null is considered a valid transformation). During source mapping you enter non-null transformations for as many target columns as needed. Each target column must have a null or non-null transformation.

To define the source mappings for a target table:

**Procedure**

Click the **Source Mapping** tab. Each target column is shown in the **Source Mapping Definition Grid**.

**Select a Data Set**

**Procedure**

Make sure the desired data set is displayed in the grid. To work with a different data set, click on the selected data set name and select a different data set from the dropdown menu.

**Note:** When you have checked all of the data sets and all of the SQL Expressions are correct, you are ready to **Review Column Analysis Data**.

By default, the first data set (as shown in order on the **Data Sets** tab) is selected for display.

In the example above, four target columns are currently mapped from table columns in the HQ data set. These target columns were created by dragging table columns into the target table. The last three target columns are null because they were created by clicking the + icon.

The **Data Sets Column Grid** lists all tables and columns in this data set. If a table column is currently mapped to a target column, the target column name is listed in **Target Columns**. This allows you to easily track which source columns have been mapped to the target table.

**Enter the Join, Where, Group By, and Having Expressions**

**Procedure**

Enter the appropriate Join, Where, Group by, and Having expressions that define the available data in the source columns of this particular data set, according to the number of source tables in this data set that contribute source columns to this target table. The target table will only contain rows that pass all of the expressions defined here.

1. **If no source columns from this data set will map to this target table, do not enter any expressions. Return to **Select a Data Set** and choose a different data set to work with.**

2. **If all of the source columns in this data set that map to this target table are from the same source table, the Join field must contain the name of the source table. The Where, Group By, and Having expressions are optional. Continue with **Map Source Columns to Target Columns**.**

   For example, if exactly five source columns in this data set map to the target table, and all five source columns come from the source table COMMUNITY_BRCH, then the Join field must contain the expression COMMUNITY_BRCH.
3. If two or more source tables in this data set contribute source columns to this target table, the **Join** field must contain a valid join condition between all source tables in this data set that contribute columns. The Where, Group By, and Having expressions are optional. Continue with **Map Source Columns to Target Columns**.

For example, if the data set contains the source tables COMMUNITY_TYPE_ACCT and COMMUNITY_BRCH, and both contribute columns to this target table, the **Join** field may contain the join condition COMMUNITY_TYPE_ACCT JOIN COMMUNITY_BRCH ON COMMUNITY_TYPE_ACCT.ACCOUNT_CODE = COMMUNITY_BRCH.ACCT_CD.

As a shortcut, enter source columns from at least two tables in this data set into the **Transformation** column and click the word **Join**. If primary PF Keys exist between the tables, Discovery automatically populates the join with that expression.

If there is no primary PF Key, or if you want to view additional options, click + to expand the **Join** field. All PF Keys and columns matches between the tables are listed. Retype or copy and paste the desired expressions into the expression editor to define the join.

- If the **Join** field already contains a valid SQL expression, clicking **Join** will not modify the expression.
- If it contains an invalid expression, it will not be automatically replaced. Delete the invalid expression and click **Join** again to populate the field.
- If the field contains a partial expression, clicking **Join** completes the expression. For example, if a three-table join is required but the expression currently contains a join between only two of the tables, clicking **Join** adds a join to the third table, if one exists.

Expressions are validated during entry. Invalid expressions are marked with a red X. Correct the expression before continuing.

### Map Source Columns to Target Columns

**About this task**

Expressions can only be based on columns from source tables that are part of this data set’s join expression. See **Enter the Join, Where, Group By, and Having Expressions**.
A target column can map from two or more source columns from the same data set. To remove a target column mapping, edit the transformation or change it to null.

**Procedure**

1. To edit a target column transformation, click in the **Transformation** field of the desired target column. The current transformation appears in the Expression Editor.

2. Enter the transformation in the Expression Editor. This can be done in two ways:
   - Manually enter the source table and column name in the Expression Editor.
   - Click on the source column in the **Data Sets Column Grid** and drag it into the Expression Editor, then edit the transformation.
The column expression now appears in the Expression Editor and in Column Expression. Also, the target column name appears in the Data Sets Column Grid and is mapped to the particular table column.

3. Continue mapping target columns from source columns in this data set until all required target columns are mapped. Then return to Select a Data Set to work in the next data set.

When you have mapped all required target columns in all data sets, you are ready to Run Unified Column Analysis.
Source Mapping Window

The Source Mapping window allows you to add more table columns to each target column, and to refine each target column with Where, GroupBy, and Having expressions. This screen is empty until you Add Columns to the Target Table Schema.

Common Window Elements describes the menus, tabs, and buttons (like Activity Viewer and Error List) that are common to all windows.

The Source Mapping Window contains the following areas:

- Target Table Schema List
- Source Mapping Definition Grid
- SQL Expression Pane
- Data Sets Column Grid
- The Expression Editor

Source Mapping Definition Grid

The Source Mapping Definition Grid displays each target column in the selected target table along with its transformation for the current data set, and its data type. Define Source Mappings in this window before running Unified Column Column Analysis.
Source Mapping Definition Grid Tools
About this task

The following standard tools are available for viewing the data in the Source Mapping Definition Grid. Also see the Grid Panes description for common actions you can perform in a grid.
The selected data set is displayed in the Data Sets Column Grid. It also filters the Source Mapping Definition Grid to display only the columns in that data set, whether or not they are used in the target table. All transformations for the target table to this data set are listed in the grid. To display transformations for a different data set, select a different data set from the dropdown menu.

- Suggest Transformations
- Data Preview
- Update Count
- Advanced Search
- Search
- Auto Filter Row
- Column Chooser
- Export Grid Content

**Source Mapping Definition Grid Columns**
The Target Table list displays data about each target table in the current project. This data is updated immediately whenever changes are made.

- Target Columns (Unified Schema)
- Transformation (Unified Schemas)
- Data Type

**Suggest Transformations**
The Suggest Transformations screen helps you map data (define transformations) from the current data set to the target table.

Various transformations are suggested based on the following criteria:

**Value overlaps**
You must have previously run the Overlaps task to have a value overlap suggestion.

Transformations suggested by value overlap are based on matches between source columns in the current data set and a hypothetical target table that aggregates rows from all already mapped data sources. The projected overlap numbers are based on two pieces of information:

- Transformations already entered for this and other data sets
- Existing overlaps between the source columns in the current data set and already-mapped source columns from the other data sets

**Column classification**
In order to have column classification suggestions, you must run the "Column Classification: Value Matching" task previously to classify column or discovered classification. Discovery suggests transformations as a column from current data source with the same classification as already mapped columns mapped to the same target column. Discovery also suggests transformations based on name equality between column name and classification label.

**Column names**
Mapping suggestions are done based on name equality or fuzzy matching between columns from currently mapped data source and columns from already mapped data sources to the same target column. Fuzzy matching precision is controlled by option 'Fuzzy matching threshold when compare names for suggested transformations.' defined for Match and Merge task. Default value: 0.6. Option defines the minimum % length of common
character sequences of the larger from the two compared strings to be considered as a match. For instance for this value 0.5 two string '1324' and '1526' will be consider as a match ('12' is the same common sequence, 2/4 = 0.5) but '1324' and '15267' will not (2/5 < 0.5)

Columns in the **Suggest Transformations** screen are listed below.

- **Target Column**
- **Table**
- **Column**
- **CDE**
- **Classification**
- **Selectivity**
- **Suggested by Overlaps**
  - Min Row Hit Rate with Target
  - Min Value Hit Rate with Target
  - Overlaps with
- **Suggested by Classification or Name**
  - Classification Matches
  - Name Matches

For more information, see **Suggest Transformations Tool Examples**

**Source Mapping - Show Details**

The **Show Details** screen appears when you click a statistic in the **Suggest Transformations** screen. It displays the statistics between the current data set’s source table and the already-mapped tables from the other data sets.

This screen is available after the Overlaps step has been run.

The columns in the Show Details screen are:

- **Data Set (Mapped Table)**
- **Table (Mapped Table)**
- **Column (Mapped Table)**
- **Table (Source Table)**
- **Column (Source Table)**
- **Row Hit Rate (Mapped)**: the number of rows in Mapped table column with the corresponding value in Source table column, divided by total number of rows in the Mapped table
- **Row Hit Rate (Source)**: the number of rows in the Source table column with the corresponding value in the Mapped table column, divided by total number of rows in the Source table
- **Value Hit Rate (Mapped)**: the number of unique values in the Mapped columns with corresponding values (hits) in the Source column, divided by the Cardinality of the Mapped column
- **Value Hit Rate (Source)**: the number of unique values in the Source with corresponding values (hits) in the Mapped column, divided by the Cardinality of the Source column
- **Selectivity - Mapped (Source Mapping Show Details)**
- **Selectivity - Source (Source Mapping Show Details)**

**Source Mapping Data Preview**

The Source Mapping Data Preview screen displays source column data from the joined source tables in the current data set along with mapped data. Because both sub-views extract data from the current data set, they share the same preview criteria.
The data preview has 2 panels: **Source Data** (top) and **Source Mapping Results** (bottom).

- **Source Data** shows the joined, unfiltered data in the current data set. All columns from joined tables are shown. In displaying the data, it uses the join expression but ignores the remaining SQL expressions (where, group by, and having) from the **SQL Expression Pane**. It also ignores transformations.

- **Source Mapping Results** displays the data extracted from the current data set using the entire SQL Expression defined on the Source mapping screen for the current data set, including where, group by, and having.

  **Source Mapping Results** has the same number of columns as the target table.

**SQL Expression Pane**

The SQL Expression field displays the join, where, group by, and having clauses that define the available data in the source columns of this particular data set. The target table will only contain rows that pass the entire SQL Expression defined here.

If at least one source column from this data set is mapped to a target column, a **Join** expression is required. The **Where**, **GroupBy**, and **Having** fields are optional.

For instructions on using this pane, see **Enter the Join, Where, Group By, and Having Expressions**.

### Perform Unified Column Analysis

#### Overview

The Unified Column Analysis screen allows you to analyze the combined (aggregated) data in each target table column. The resulting statistics and previews are used to set up and perform Match and Merge analysis.

Before you [Run Unified Column Analysis](#) or [Re-run Unified Column Analysis](#) all target columns should be defined and the source mappings should be complete, as described in **Define a Target Table Schema** and **Define Source Mappings**.

#### Run Unified Column Analysis

**About this task**

Before running or re-running Unified Column Analysis, all target columns should be defined and the source mappings should be complete. See the **Overview**.
To perform Unified Column Analysis processing:

**Procedure**

1. Click **Run Next Steps** in the *Source Mapping* window. The *Processing Options* dialog appears with **Unified Column Analysis** selected.

   **Note:** The Unified Column Analysis step has no processing options. Processing is performed on all columns in the target table.

2. If matching conditions have been entered in the *Match and Merge Analysis* window, you can drag down the **Steps** arrow to **Run Match and Merge Analysis** at the same time.

3. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   **Note:** Modifying task options is an advanced activity. Options are described in the Reference of Task Options. Modifying options and creating new option sets is described in Option Sets.

4. Click **Run** to execute Unified Column Analysis processing. If the procedure does not complete correctly, see Troubleshooting Task Execution.

   **Note:** Although the Unified Column Analysis window is displayed immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.
Re-run Unified Column Analysis

The Re-run Step button allows you to perform Unified Column Analysis from within the Unified Column Analysis screen. This updates the statistics in all Unified Schema tabs.

About this task

Before running or re-running Unified Column Analysis, all target columns should be defined and the source mappings should be complete, as described in Define a Target Table Schema and Define Source Mappings.

Use Re-run Step in the Unified Column Analysis window when:
- You make any changes to a target column in the Target Table Schema Window.
- You change any transformations in the Source Mapping Definition Grid.
- You click on the Modify Physical Tables in the Data Set button. Drop the staging tables before re-running this step.
- You clicked directly on the Unified Column Analysis tab after refining columns in the Source Mapping Window instead of using Run Next Steps.

To re-run Unified Column Analysis processing:

Procedure

1. Click Re-run Step... in the Unified Column Analysis window. The Processing Options Dialog appears with Unified Column Analysis selected.

Note: The Re-run Step function has no processing options. Processing is performed on all columns in the target table.
2. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

    **Note:** Modifying task options is an advanced activity. Options are described in [Reference of Task Options](#). Modifying options and creating new option sets is described in [Option Sets](#).

3. Click **Run**.

    **Note:** Although the Unified Column Analysis screen is displayed immediately, it may take some time for the actual data processing to complete. Check the [Activity Viewer](#) for the status of the task.

### Review the Unified Column Analysis Results

**About this task**

*After running Unified Column Analysis, review the resulting statistics in the [Unified Column Analysis Grid](#). Statistics are shown for the aggregated columns in the target table, as well as for each individual table column used in the target table.*

As needed, refine the target table by returning to the Target Table Schema window and adding or removing target table columns. You may also change the source mapping.

*After making any changes, [Re-run Unified Column Analysis](#) to refresh the statistics, then review the results again.*

### Unified Column Analysis Window

*This screen is empty until Unified Column Analysis has been run.*

**About this task**

The Unified Column Analysis tab shows the **statistics for all aggregated columns** in this target table. All grids are empty until you **Define a Target Table Schema**.

Click a column’s + button to display all transformations defined for this column.

[Common Window Elements](#) describes the menus, tabs, and buttons (like **Activity Viewer** and **Error List**) that are common to all windows.
The Unified Column Analysis screen contains the following areas:

- **Target Table Schema List**
- **Unified Column Analysis Grid**

**Unified Column Analysis Grid**

The Unified Column Analysis grid displays statistical information for each target table column, and provides several tools for analyzing the quality of the data in each column.
Unified Column Analysis Tools

The following tools are available in the Unified Column Analysis grid. Also see the Grid Panes description for common actions you can perform in a grid.

- Advanced Search
- Search
- Auto Filter Row
- Column Chooser
- Export Grid Content
- Unified Column Analysis Data Preview
- Frequency (Value, Pattern, Length)
- All Mappings
- Export Table

Export Grid Content

About this task

This function exports the metadata in the target table rows only (Target Table band and Target Table Statistics band). To export the actual data in the Mapping Band and Mapping Statistics Band, display and export the All Mappings grid.

Unified Column Analysis Data Preview

The Preview Data button displays the current contents of the target table. This is a read-only view that allows you to sort and filter the target table data.

You can use the information in this view to identify which data in the target table requires analysis.
Frequency (Value, Pattern, Length)

The following views list the frequency of values, patterns, and lengths in the selected target table columns, along with the transformations from each mapped data set to these columns.

- **Value Frequency**
- **Pattern Frequency**
- **Length Frequency**

**Value Frequency:**

Value Frequency lists each value in the selected target table column, and displays the number of occurrences of that value in each data set. The default display lists the most frequent value first.

**About this task**

For example, in the following screen shot, the target table column CITY was selected and then the Value Frequency button was clicked. The Value Frequency window displays each value in the aggregated CITY target table column and indicates how often each value appears in each data set and in this target column.

In this example, the value SEATTLE appears 14 times in this column in the target table, 11 times in the mapped column from the HQ data set, twice in West, and once in Northwind.
The Value Frequency window contains a Details button. This allows you to select a particular value and view all rows in the aggregated target table containing this value. The data set that is the source of each row in the target table is also shown.

In the example above, the value TACOMA is in focus. The lower portion of the window displays information about all 13 instances of that value in the target table column, including 9 instances in the HQ data set and 4 instances in West.

**Pattern Frequency:**

The Pattern Frequency tool allows you to view the value patterns in the target table column and all data set transformations.

**About this task**

In the following screen shot, the target table column POSTAL_CODE was selected and then the Pattern Frequency button was clicked. The Pattern Frequency window displays each varchar(30) pattern in the aggregated POSTAL_CODE target table column and indicates how often each pattern appears in each data set and in this target column.

In this example, the pattern NNNNN appears 88 times in this column in the target table: 32 times from HQ, 6 times from West, and 50 times from Northwind.
The Pattern Frequency window contains a Details button. This allows you to select a particular value and view all rows in the aggregated target table containing this value. The data set that is the source of each row in the target table is also shown.

In the example above, the value NNNN-NNN is in focus. The lower portion of the window displays information about the nine instances of that value in the target table column. All occurrences of this value appear in postal codes from Brazil.

**Length Frequency:**

The Length Frequency tool allows you to view the length of the values in the target table column and all data set transformations.

**About this task**

In the following screen shot, the target table column EMPLOYEE_ID was selected and then the Length Frequency button was clicked. The Length Frequency window displays each value length in the aggregated EMPLOYEE_ID target table column and indicates how often each length appears in each data set and in this target column.

In this example, a length of 6 characters appears 23 times in this column in the target table, all from the HQ data set.
The Length Frequency window contains a Details button. This allows you to select a particular value and view all rows in the aggregated target table containing this value. The data set that is the source of each row in the target table is also shown.

In the example above, the length of 5 is in focus. The lower portion of the window displays information about the nine instances of that length in the target table column. The EMPLOYEE_ID column in the lower portion of the window contains nine rows, with each value having exactly five characters.

**All Mappings**

The All Mappings button displays a summary list with each target column, the datatype of the target column, and the transformation for that target column for each data set. Some or all rows in this grid can be exported in a variety of formats using the Export Grid Content button.
Export Table

The data in the target table can be exported to a new table in the staging data source by clicking Export Table. The exported table has an extra column, SDD_DATA_SET, containing the name of the source data set.

When the target table is exported in this manner, the staging table created by Discovery uses the following staging-specific date-time data types.

<table>
<thead>
<tr>
<th>Table 14. Date-time data types after table is exported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Staging Database</td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>DB2</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Table 14. Date-time data types after table is exported (continued)

<table>
<thead>
<tr>
<th>Staging Database</th>
<th>Target Table Data Type</th>
<th>Exported Staging Table Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>Date</td>
<td>Date</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>Timestamp</td>
</tr>
<tr>
<td></td>
<td>DateTime</td>
<td>Date</td>
</tr>
<tr>
<td></td>
<td>Timestamp</td>
<td>Timestamp</td>
</tr>
<tr>
<td></td>
<td>Datetimestring</td>
<td>Timestamp</td>
</tr>
</tbody>
</table>

Unified Column Analysis Statistics

Click a target table column's + icon to expand the view and display the Column Expression for each table column mapped to that target column, along with statistics.

Target Table band:
- Name
- Data Type

Target Table Statistics band:
- Mapped Data Sets (Unified Schema)
- Cardinality
- Selectivity
- Min
- Max
- Null Count (available from Column Chooser)
- Non-Null Rows (available from Column Chooser)
- Non-null Selectivity (available from Column Chooser)

When a column in the Unified Column Analysis grid is expanded, the following information is displayed about the column:

Mapping Band:
- Data Sets (Unified Schema)
- Transformation (Unified Schemas)

Mapping Statistics Band:
- Cardinality
- Selectivity
- Min
- Max
- Completeness Rate

Perform Match and Merge Analysis

Overview

All work in this screen is performed on the target table schema. No matching or merging is performed on any of the source tables.
The *Match* task splits the aggregated target table records into groups containing matching (duplicate) records. The *Merge* task resolves conflicts to produce a single record for the group and creates the unified target table.

**Matching**
The Match task splits the aggregated target table records into groups that contain matching (duplicate) records. Two records from the target table match if they satisfy the matching condition. Records belong to the same group if they meet both of the following criteria:

- For each record in the group, another matching record exists in the same group.
- No records in the group match any records in any other group.

The Statistics Generated During the Match Step reflect the quality of the Matching Condition.

Automatically-generated Conflict Detection Rules for each target column determine if data from this column of duplicate records have a conflict. You can edit the conflict detection rules to fine-tune them for each target column.

The Match and Merge Analysis Groups Data Preview displays the data in each group and allows you to view certain types of groups, such as groups with discrepancies or groups without source duplicates.

**Matching Condition:**
A matching condition is a SQL condition used to compare each row in the aggregated target table against every other row. One matching condition is selected as the primary matching condition for the table.

**Pseudo Names for the Target Table in SQL Expressions:**
In the Match and Merge step, matching conditions are used to match each row against every other row. Pseudo names are used in the matching condition expression to represent different rows from the target table: `DM_ROW1` and `DM_ROW2`. These names represent two different rows from the target table and are used for the target table rows matched against each other.

For instance, to group records (duplicates) by the same SSN columns, the SQL expression is `DM_ROW1.SSN = DM_ROW2.SSN`. This means “the SSN of each row should match the SSN of every other row.” These same pseudo names are used for all expressions entered in the Match and Merge screen.

**Creating Groups:**
During the Match task, Discovery creates groups.

**Procedure**
1. In the initial grouping, Discovery creates groups containing rows that satisfy the matching condition. Each group is given a unique number for tracking purposes.
2. If the same row is used in two different groups, those groups are combined in a single group (transitive closure).
3. An exclusive group is added so that all records can be grouped.
Results

This process guarantees that no records in a group match any record from other groups.

Group Numbers:

Each combined group has a unique number, but the numbers may not be sequential.

The actual number of each group is not important; it is just an internal Discovery tracking number that identifies a particular group of records as being distinct from some other group of records. If the matching condition is modified and the task is run again, the group numbers are likely to change.

Transitive Closure:

When the matching condition contains only the following types of conditions, Discovery does not perform transitive closure because these conditions do not produce transitive matches:
- conjunction between predicates
- equality as a predicate conditions
- simple operations (+, -, /, *, ||) between columns of the target table (no functions),
- constants

For all other matching conditions, Discovery performs transitive closure. This may require multiple steps and may be time consuming or generate undesirable results.

You can control the number of transitive closures steps performed after the initial matching with the **Number of transitive closure iterations** task option in the **Match and Merge Analysis** group. See [Run Match and Merge Analysis](#).

Example:

As an example, assume that employee records are considered duplicates (belonging to the same group) if the following matching condition returns true. Note that **DM_ROW1** and **DM_ROW2** are consistently used to identify different rows from the target table.

```sql
DM_ROW1.FirstName = DM_ROW2.FirstName AND DM_ROW1.LastName = DM_ROW2.LastName OR DM_ROW1.EmployeeID = DM_ROW2.EmployeeID
```

<table>
<thead>
<tr>
<th>Row id</th>
<th>EmployeeId</th>
<th>FirstName</th>
<th>LastName</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>10023</td>
<td>Smith</td>
<td>Lindon</td>
</tr>
<tr>
<td>R2</td>
<td>10024</td>
<td>Greg</td>
<td>Lindon</td>
</tr>
<tr>
<td>R3</td>
<td>10023</td>
<td>Ray</td>
<td>Eagle</td>
</tr>
<tr>
<td>R4</td>
<td>10025</td>
<td>Martha</td>
<td>Steward</td>
</tr>
<tr>
<td>R5</td>
<td></td>
<td>Smith</td>
<td>Lindon</td>
</tr>
<tr>
<td>R6</td>
<td></td>
<td>Martha</td>
<td>Steward</td>
</tr>
</tbody>
</table>
The first step of the Match task creates the following initial grouping. Note that R3 and R5 do not satisfy Matching Conditions.

Table 16. Match task

<table>
<thead>
<tr>
<th>GROUP_NUMBER</th>
<th>DM_ROW1.Row id</th>
<th>DM_ROW2.Row id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R1</td>
<td>R3</td>
</tr>
<tr>
<td>2</td>
<td>R1</td>
<td>R5</td>
</tr>
<tr>
<td>3</td>
<td>R4</td>
<td>R6</td>
</tr>
</tbody>
</table>

After the second step in the Match task, transitive closure, the following groups remain. Note that R3 and R5 belong to the same group because they both match R1.

Table 17. Groups that are left after the second step

<table>
<thead>
<tr>
<th>GROUP_NUMBER</th>
<th>DM_ROW1.Row id</th>
<th>DM_ROW1.Row id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R1</td>
<td>R3</td>
</tr>
<tr>
<td></td>
<td>R1</td>
<td>R5</td>
</tr>
<tr>
<td>3</td>
<td>R4</td>
<td>R6</td>
</tr>
</tbody>
</table>

In order to have all target table records grouped, an exclusive group is added with records that do not match any other records.

Table 18. Adding an exclusive group

<table>
<thead>
<tr>
<th>GROUP_NUMBER</th>
<th>DM_ROW1.Row id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R1</td>
</tr>
<tr>
<td>1</td>
<td>R3</td>
</tr>
<tr>
<td>1</td>
<td>R5</td>
</tr>
<tr>
<td>3</td>
<td>R4</td>
</tr>
<tr>
<td>3</td>
<td>R6</td>
</tr>
<tr>
<td>4</td>
<td>R2</td>
</tr>
</tbody>
</table>

Conflict Detection Rules:
About this task

To display the Conflict Detection Rules, click in the Transformation Statistics Band and display the Column Chooser, then select Conflict Detection Rule.

A conflict detection rule is a SQL condition automatically created by Discovery when the target column is created, to determine how consistent the column values are within a group. Each target column has its own conflict detection rule used to detect conflicts for this target column between rows in the group.

During the Match task, Discovery applies the conflict detection rule to each group in each column, for all combinations of different rows in each group. If the conflict detection rule returns true, then two values from these records for this column are considered consistent (no conflicts detected).

Various statistics are calculated that reflect the consistency (quality) of the matching conditions. Review the Statistics Generated During the Match Step and use data preview to determine if the conflict detection rules are accurate. Modify
them, if needed, if needed, to correctly represent the quality of the data in the group.

Default Conflict Detection Rule:

About this task

The default conflict detection rule is shown below. It means that if the values for the column for different rows of the same group are equal, or if one of the values is null, there is no conflict.

\[
\text{DM\_ROW1.\text{column}} = \text{DM\_ROW2.\text{column}} \lor \text{DM\_ROW1.\text{column}} \text{ IS NULL} \lor \text{DM\_ROW2.\text{column}} \text{ IS NULL}
\]

You can restore the default conflict detection rule by deleting the current conflict detection rule, then re-running the Match step or closing and re-opening the project.

Discovery uses conflict detection rules to discover the consistency of the matching conditions and which source of data (data set) is trustworthy. Consistency and trustworthiness are expressed in calculated Trust Level and Precedence fields on the Transformation level of the Match and Merge grid.

More consistent data produces a higher Trust level (up to 1.0) and lower Precedence.

Example:

Data is consistent if it has few conflicts with other rows from the same group. For instance, look at the following group.

<table>
<thead>
<tr>
<th>GROUP_NUMBER</th>
<th>SDD_DATA_SET_</th>
<th>FirstName</th>
<th>LastName</th>
</tr>
</thead>
<tbody>
<tr>
<td>840</td>
<td>Data Set 1</td>
<td>Greg</td>
<td>Olson</td>
</tr>
<tr>
<td>840</td>
<td>Data Set 2</td>
<td>Greg</td>
<td>Olson</td>
</tr>
<tr>
<td>840</td>
<td>Data Set 3</td>
<td>Greg</td>
<td>Olsen</td>
</tr>
</tbody>
</table>

There is a conflict between the values Olson and Olsen (in Data Set 3). As a result, the default conflict detection rule returns false. Data Set 1 and Data Set 2 are considered more trustworthy because there are no conflicts between records from these data sets and they each have one conflict with Data Set 3. Data Set 3 is considered less trustworthy because the default conflict detection rule fails twice for it: once for the comparison with Data Set 1, and a second time for the comparison with Data Set 2.

The trust level of the transformation for all columns is accumulated in a Confidence Index for the matching condition. This is an important statistic that reflects the quality (consistency) of the matching condition. A higher confidence index (up to a maximum of 1.0) indicates a higher quality matching condition.

Statistics Generated During the Match Step:

During the Match task execution, the following statistics are generated:
About this task

- Group statistics (in Match and Merge Analysis Groups Data Preview)
- Matching condition statistics (in Match and Merge Statistics Band)
- Target column statistics (in Target Column Statistics Band)
- Transformation statistics (in Transformation Statistics Band)

Review these statistics and use the Match and Merge Analysis Groups Data Preview to determine the quality of each group (target column) and the consistency of the data in each group. This also allows you to analyze and compare alternative matching conditions for each target column.

Group and Matching statistics information persists in the metadata and in the following staging tables:

GROUPS_<unique id>:
A grouping table that stores the references to all records. Records are associated with the unique group number (GROUP_NUMBER) field of the table

MATCH_<unique id>:
A match table that stores the statistics or matching attributes discovered or calculated during the Match step.

Precedence:

About this task

During the Match task, Discovery also assigns a Precedence value for each transformation in each data set based on the discovered statistics. More trustworthy transformations (with a higher Trust Level) have a lower precedence value.

If precedence is not discovered for a data set, or the precedence is empty, the corresponding transformation is considered less trustworthy than other transformations to which Discovery did assign precedence values.

You can edit the precedence value for each transformation, if needed. The precedence is re-discovered every time the Match step is run, unless the corresponding target column is marked as Approved.

Precedence is an important value in the Merge process. Precedence is also used in Conflict Resolution Rules.

Precedence is discovered during the Match task. You can use the precedence in the conflict resolution rule to resolve the conflict for two rows of the target table. To do this, use one of the following syntax lines in the conflict resolution rule:

```sql
Precedence (DM_ROW1.SDD_DATA_SET_, DM_ROW1.<column_name>)
```

or

```sql
Precedence (DM_ROW2.SDD_DATA_SET_, DM_ROW2.<column_name>)
```

Precedence is only useful in the context of two data sets. If Merge is done for one data set only, the resulting precedence values are meaningless.
Merging

About this task

After verifying that the groups are correct and reviewing the Precedence of each data set, run the Merge task. This creates a single record for each group.

The Merge task selects single values from multiple alternatives. During the Merge process, Discovery uses conflict resolution rules.

Review the Merge results in the Target Column Statistics Band and the Column Conflicts Data Preview. If needed, modify the Conflict Resolution Rules for one or more target columns and re-run the Merge task.

All merged records are placed in the staging table MERGE_<unique id>.

Conflicts Resolution Rules:

About this task

To display the Conflict Resolution Rules, click in the Target Column Statistics Band and display the Column Chooser, then select Conflict Resolution Rule.

Conflict resolution rules are SQL expressions that compare two rows in a group and determine which value is the preferred one.

As each target column is created, Discovery automatically creates a conflict resolution rule for it. If this conflict resolution rule does not select the preferred value in the merged record for a column, you can edit the conflict resolution rule and re-run the Merge task.

In the Match process, a conflict resolution rule is used to compare two values from different rows in the same group. For each target column, the value in the row that wins the rule most often during a row comparison is selected as the preferred value for that group during the Merge step.

Important information about the conflict resolution rule:
- It is a valid SQL expressions with target table rows.
- You can edit each column’s conflict resolution rule to meet specific merging criteria.
- The conflict resolution rule can use columns other than the target column for which the rule is defined.
- Using the Precedence value is helpful but is not necessary; the conflict resolution rule can be any valid SQL expression that uses target table columns.

Example 1: Default precedence:

The following example illustrates the usage of discovered precedence for the Merge step in the default conflict detection rule.

The following All Groups data preview shows data from three different data sets.
In this data preview, you can see the results of grouping using the matching condition DM_ROW1.SSN = DM_ROW2.SNN (grouping records with the same social security number). The default conflict resolution rules were used, which detected discrepancies for the FIRSTNAME column during the Match step. As shown below, the FIRSTNAME column does not have complete confidence for all transformations. The maximum Trust Level is 1.0.
The Trust Level indicates that a higher trust level is assigned to the transformation from Data Set 1. As seen in the first screen shot in this example, Data Set 3 has more conflicts than the other two data sets, and Data Set 2 has an exclusive group, which makes it less trustworthy than Data Set 1.

As a result, the default conflict resolution rule is modified to use the transformation with the higher precedence. The following CASE statement selects the value 'JOHN' over the value 'JONATHAN' because the WHEN condition is true for rows with 'JOHN' and not true for rows with 'JONATHAN'.

```
CASE WHEN DM_ROW1.FIRSTNAME IS NULL  THEN DM_ROW2.FIRSTNAME  WHEN DM_ROW2.FIRSTNAME IS NULL  THEN DM_ROW1.FIRSTNAME  WHEN Precedence(DM_ROW1.SDD_DATA_SET , DM_ROW1.FIRSTNAME ) < Precedence(DM_ROW2.SDD_DATA_SET , DM_ROW2.FIRSTNAME )  THEN DM_ROW1.FIRSTNAME  ELSE DM_ROW2.FIRSTNAME  END
```

The results of the Merge step using the modified conflict resolution rule are shown below. The value 'JOHN' has been selected as the preferred value.
Example 2: Customized Conflict Resolution:

This is an example of editing the default conflict resolution rule.

Consider the following data grouped by the column WRHS_ITEM_ID.
The goal is to merge the record using:

- Longest ITEM_DESCR
- Highest level of ITEMS_REMAINING
- Latest LAST_SHIPPED date

During the Match step, all precedence values returned 1 because there is only one data set.

Edit the conflict resolution rule as follows before running Merge:

- Longest ITEM_DESCR: `CASE WHEN length(DM_ROW1.ITEM_DESCR) > length(DM_ROW2.ITEM_DESCR) THEN DM_ROW1.ITEM_DESCR ELSE DM_ROW2.ITEM_DESCR END`
- Highest level of ITEMS_REMAINING: `CASE WHEN DM_ROW1.ITEMS_REMAINING > DM_ROW2.ITEMS_REMAINING THEN DM_ROW1.ITEMS_REMAINING ELSE DM_ROW2.ITEMS_REMAINING END`
- Latest LAST_SHIPPED: `CASE WHEN DM_ROW1.LAST_SHIPPED > DM_ROW2.LAST_SHIPPED THEN DM_ROW1.LAST_SHIPPED ELSE DM_ROW2.LAST_SHIPPED END`
The Merge step uses this expression to compose one record for each group, shown below. Note that data in these records are taken from different records in each group.
Default Conflict Resolution Rule:

About this task

The default conflict resolution rule for a target column is shown below.

```sql
CASE WHEN DM_ROW1.<column> IS NULL
THEN DM_ROW2.<column> WHEN DM_ROW2.<column> IS NULL
THEN DM_ROW1.<column> WHEN Precedence( DM_ROW1.SDD_DATA_SET_ ,
DM_ROW1.<column> ) < Precedence( DM_ROW2.SDD_DATA_SET_ ,
DM_ROW2.<column> ) THEN DM_ROW1.<column>
ELSE DM_ROW2.<column> END
```

This is read as "given two compared values, where one value is null, the non-null value is preferred; and if both values are non-null, the value with the smaller precedence is preferred."

You can restore the default conflict resolution rule by deleting the current conflict resolution rule, then re-running the Merge step or closing and re-opening the project.

Run Match and Merge Analysis

About this task

Before running or re-running Match and Merge Analysis, you need to run `Run Unified Column Analysis`.

You also need to define at least one matching condition using the `Define Matching Conditions` procedure. This task uses the primary matching condition.

To perform Match and Merge Analysis processing:
Procedure

1. Click Run Next Steps in the Unified Column Analysis window. The Processing Options Dialog appears with Match and Merge Analysis selected.

2. Verify that the task Options are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   **Note:** Modifying task options is an advanced activity. Options are described in Reference of Task Options. Modifying options and creating new option sets is described in Option Sets.

3. Click Run to execute Match and Merge Analysis processing. Processing is performed using the primary matching condition on all columns in the target table. If the procedure does not complete correctly, see Troubleshooting Task Execution.

   **Note:** Although the Match and Merge Analysis window is displayed immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.

Re-run Match and Merge Analysis

About this task

Before re-running Match and Merge Analysis, you need to define at least one matching condition using the Define Matching Conditions procedure. This task uses the primary matching condition.
The **Re-run Step** button allows you to perform Match and Merge Analysis from within the Match and Merge Analysis screen.

Re-run the Match task when:
- You define or modify a matching condition.
- You modify a conflict detection rule.
- You run **Run Unified Column Analysis** or **Re-run Unified Column Analysis**.

Re-run the Match and Merge tasks when:
- You modify a conflict resolution rule.
- You have finished defining the matching conditions and are ready to perform the Merge task.

(The Match task must be selected in order to run the Merge task.)

To re-run Match and Merge Analysis processing:

**Procedure**
1. Click **Re-run Step** in the **Match and Merge Analysis** window. The Processing Options Dialog appears with **Match and Merge Analysis** selected.
2. Select the appropriate options. By default, both options are selected.
   - To run both Match and Merge processing, leave both **Match** and **Merge** selected.
To run Match processing, use the **Match** option only. This updates all of the matching condition statistics.

To run Merge processing, use the **Merge** option only. If the matching condition is complex, de-selecting this option may save processing time. After refining the matching condition, if necessary, you can run Merge as a separate step.

**Note:** Processing is performed using the primary matching condition on all columns in the target table.

3. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

**Note:** Modifying task options is an advanced activity. Options are described in Reference of Task Options. Modifying options and creating new option sets is described in Option Sets.

4. Click **Run**.

**Note:** Although the Match and Merge Analysis screen is displayed immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.

### Define Matching Conditions

#### About this task

After you have run Unified Column Analysis, you are ready to define matching conditions for the target table.

Before defining matching conditions, read **Matching**

To define matching conditions:

**Procedure**

Click the **Match and Merge Analysis** tab.

**Add a Matching Condition**

**Procedure**

1. Click + to add a new, empty **Matching Condition**

2. Click in the **Matching Condition** field and enter a matching condition in The Expression Editor. For more information on matching conditions, see Pseudo Names for the Target Table in SQL Expressions.
Run Matching Condition Analysis
Procedure

Run the Match task as described in [Run Match and Merge Analysis](#).

**Note:** The Match and Merge task uses the primary matching condition. To compare an alternative matching condition, mark the alternative matching condition as primary and re-run the Match task.

Review Matching Condition Analysis
Procedure

1. When Match processing is finished, review the statistics in the Match and Merge Analysis Grid. The Merge Summary presents the same statistics in an alternative view. Also preview the actual results using the Match and Merge Analysis Groups Data Preview and review the Precedence.

2. As needed, refine the matching conditions or Conflict Detection Rules and Re-run Match and Merge Analysis running only the Match task, until the results are satisfactory. This includes:
   - minimal Conflicts Rate
   - high Group Consistency Rate
   - high Confidence Index
   - correct Precedence

3. Re-run Match and Merge Analysis running both the Match and the Merge tasks.

4. Review the results in the data previews, including the Column Conflicts Data Preview. As needed, refine the Conflict Resolution Rules and re-run the Match task until the Conflicts Rate is acceptable.

Export Match and Merge Results

The data in the Match and Merge Analysis tab can be exported. Data is exported in the datatypes defined in the Target Table Schema tab.

About this task

**Note:** Discovery may convert date-time data types to staging-specific formats. For more information, see [Export Table](#).
• All or selected rows in the Match and Merge Analysis Grid can be exported in a variety of formats by clicking the Export Grid Content icon in the tool bar of the grid.

• All or selected rows in the Match and Merge Analysis Groups Data Preview screens can be exported in a variety of formats by clicking the Export Grid Content in any of the Groups Data Preview screens.

• The current Match or Merge results can be exported by selecting Export Grouped, Export Merged in the tool bar of the Match and Merge Analysis Grid. A new table is created in the repository containing all matched or merged records from the aggregated target table.

• All or selected rows in the Column Conflicts Data Preview can be exported in a variety of formats by clicking the Export Grid Content icon in the tool bar of the Conflicts Data Preview screen.

**Match and Merge Analysis Window**

The Match and Merge Analysis window shows the current matching conditions you have created for the target table, along with discovered statistics.

**About this task**

**Note:** Refresh the statistics on this screen and in the Target Table Schema Window by re-running the Match task after making any of the following changes:

• creating or modifying a matching condition
• modifying a conflict detection rule
• modifying a conflict resolution rule

Common Window Elements describes the menus, tabs, and buttons (like Activity Viewer and Error List) that are common to all windows.

**Match and Merge Analysis Grid**

**About this task**

After running the Match task, the Match and Merge Analysis Grid contains the discovered statistics for each matching condition. The primary matching condition is indicated by the selection in the Primary column. You can select a different matching condition as primary by clicking a different button in the Primary column.

If you change the target table definition or any source data, Re-run Match and Merge Analysis (Match task only) to refresh the statistics.

**Match and Merge Analysis Grid Tools**

The following standard tools are available for viewing and manipulating the data in the Match and Merge Analysis Grid. Also see the Grid Panes description for common actions you can perform in a grid.

• Advanced Search
• Search
• Auto Filter Row
• Column Chooser
• Export Grid Content
• Add a Matching Condition
- Delete a Matching Condition
- Match and Merge Analysis Groups Data Preview
- Export Grouped and Export Merged
- Merge Summary

**Match and Merge Analysis Grid Information**

**About this task**

Matching Condition statistics reflects the quality of the Matching Condition. The Merge Summary shows the same statistics in a different format.

Statistics are grouped into the following results:

- Target Table Band
- Match and Merge Statistics Band
- Target Column Statistics Band
- Transformation Statistics Band

**Target Table Band:**

- Primary
- Approved
- Matching Condition

**Match and Merge Statistics Band:**

**About this task**

This group of statistics indicates the general quality of the matching condition. To display the additional statistics, select any row in the Matching Condition Statistics Band and click the Column Chooser.

- Group Count
- De-duplication Rate
- Group Consistency Rate (available from Column Chooser)
- Confidence Index
- Column Conflict Rate (available from Column Chooser)
- Unmatched Row Rate
- Average Group Size
- Max Group Size
- Notes

Target Column Statistics Band:

Expanding the matching condition displays information about the matching condition for each target column.

About this task

To display the additional statistics, select any row the in the Target Column Statistics Band and click the Column Chooser.
- Approved
- Name
- Conflicts Rate (clicking a Conflicts Rate value displays the Column Conflicts Data Preview)
- Conflict Detection Rule (available from Column Chooser)
- Conflict Resolution Rule (available from Column Chooser)
- Notes

Transformation Statistics Band:

About this task

Expanding a target column displays information about each transformation in this target column for each data set. To display the additional statistics, select any row the in the Transformation Statistics Band and click the Column Chooser. Also use the Column Chooser to display the Conflict Detection Rule and Conflict Resolution Rule columns, which are not shown by default.
- Data Set
- Transformation (Unified Schemas)
- Trust Level
- Completeness Rate
- Precedence
- Notes

Column Conflicts Data Preview

About this task

This screen is displayed by clicking a value in the Conflicts Rate column.

The Column Conflicts Data Preview is similar to the Match and Merge Analysis Groups Data Preview but it focuses on a single target column's conflicts. To view details, select a row in the upper pane and click the Details button.
The following standard tools are available for viewing and manipulating the data in the Column Conflicts Data Preview. Also see the Grid Panes description for common actions you can perform in a grid.

- Advanced Search
- Search
- Auto Filter Row
- Details
- Column Chooser
- Export Grid Content
- Preview Criteria

The Focus on Column: Details pane is populated when you click the Details button or double-click a row in the upper grid. It displays information about the selected target column: the data from each logical column in the group, along with the merged data in the unified schema.

**Match and Merge Analysis Groups Data Preview**

After processing a matching condition, you can drill down into the results using the different Groups previews. The different views in this menu are created by using different pre-defined preview criteria. You can display the same views by manually setting the preview criteria.

After running the Match task, the data previews display grouped data. After running the Merge task, the data previews display merged data.
The available data previews are listed below.

- **All Groups** displays all groups in an unfiltered view.
- **Consistent Groups** displays all groups without conflicts, meaning the conflict detection rule is calculated as true for all columns in these groups.
- **Groups with Discrepancies** displays all groups with at least one discrepancy, meaning the conflict detection rule is calculated as false for at least one column in each of these groups.
- **Exclusive Groups** displays all groups that contain exactly one row.
- **Groups with Source Duplicates** displays all groups with more than one duplicate record from the same source.
- **Groups without Source Duplicates** displays all groups with no duplicate records from the same source.

Each data preview displays the duplicate rows (after running Matches) or the merged rows (after running Merge), along with group statistics. You can select any record and click **Details** to view grouped data associated with focused row. If Merge processing has been performed, you can **Focus** on each row to see the merged master record that Discovery created for that row, using the default **Conflict Resolution Rules**.
Review groups with particular properties to determine whether the matching condition is appropriate. For example, if the matching condition should yield groups with one or two records each, but many of the resulting groups have three or more records, you can examine the data using the Details button to determine if the data is bad, or if the matching condition needs to be modified.

The following standard tools are available for viewing and manipulating the data in the Groups data previews. Also see the Grid Panes description for common actions you can perform in a grid.

- Advanced Search
- Search
- Auto Filter Row
- Details
- Column Chooser
- Export Grid Content
- Select Preview Criteria

In the Preview Criteria screen, when you define a Where clause or Order By, you can use columns from three different sources (pseudo tables):

- MATCH_TABLE: contains group level statistics values. Generated during the Match task. The column names are:
- COLUMN_DISCREPANCIES: contains the number of discrepancies for each target column
- <target table name>: contains values from target table

The Groups data previews display group-level statistics for each matching condition. These fields persist the group level statistics in the MATCH_<unique id> staging table.

- Group Number
• Data Set (Match and Merge Analysis)
• Group Size
• Conflict Count
• Columns with Conflict
• Sources Contributed
• Max Rows from Source
• Conflict Count for <column>

**Export Grouped and Export Merged**

These two options allow you to export the grouped or merged result of Matching or Merging along with group statistics and record identity. The data is exported to a new table in the staging data source. See Export Table for instructions.

For other types of exports available in Match and Merge Analysis, see Export Match and Merge Results

**Merge Summary**

The Merge Summary button opens the Merge Summary window. This window displays, for a single matching condition, the target column statistics and the transformation statistics for all target columns in the target table.

The Merge Summary window displays the selected matching condition in the title bar. For each target column in the target table, the window displays statistics from the Target Column Statistics Band and the Transformation Statistics Band.

**Note:** Some of the columns shown by default in the Merge Summary window are hidden by default in the Match & Merge Analysis Grid.
The following columns are shown in this window:

- **Column**
- **Conflict Detection Rule** (available from Column Chooser)
- **Conflict Resolution Rule** (available from Column Chooser)
- **Conflicts Rate** (clicking a **Conflicts Rate** value displays the **Column Conflicts Data Preview**)
- **Data Set**
- **Transformation (Unified Schemas)**
- **Precedence**
- **Trust Level**
- **Completeness Rate**
- **Null Count**

The following standard tools are available for viewing and manipulating the data in the Merge Summary window. Also see the **Grid Panes** description for common actions you can perform in a grid.

- **Advanced Search**
- **Search**
- **Auto Filter Row**
- **Column Chooser**
- **Export Grid Content**

**Define Unified Schema PF Keys**

**Overview**

After prototyping target tables, you can continue unified schema analysis by analyzing PF keys within the target tables. If a target table is based on an imported CWM file containing PF Keys, you can work with those PF Keys. Alternatively, you can manually enter PF Keys on the Unified Schema PF Keys tab.
Discovery can calculate statistics for manually-defined PF Keys as well as for PF Keys in an imported CWM file. You can also preview the data.

**Note:** Discovery does not automatically discover primary-foreign key relationships for target tables.

After defining Unified Schema PF Keys or importing a CWM file containing Unified Schema PF Keys, the Unified Schema PF Keys are immediately visible in the [Unified Schema PF Keys Window](#). However, you cannot refresh the Unified Schema PF Keys statistics until you have completed source mapping, Unified Column Analysis, and Match and Merge.

**How to Use Unified Schema PF Keys**

In order to use Unified Schema PF Keys, at least two target tables must exist, with at least one target column each.

**About this task**

Unified Schema PF Keys are created in two ways:

- After manually defining target tables and completing all steps in the Unified Schema tab, you manually define Unified Schema PF Keys.
- **Import a CWM File as a Target Table Schema** where the CWM file contains PF Keys. Complete all steps in the Unified Schema tab and then display the Unified Schema PF Keys tab. The imported keys are displayed. (In addition to the imported PF Keys, you can manually define additional Unified Schema PF Keys.)

Completing all of the steps in the Unified Schema tab populates the target tables with the consolidated data. You can now refresh Unified Schema PF Key Statistics and use the statistics and data previews to determine the quality of the relationships.

The general workflow on this screen is:

**Procedure**

1. Define Unified Schema PF Keys if they were not imported.
2. Refresh Unified Schema PF Key Statistics
3. Review Unified Schema PF Keys using Show Hits, Show Misses and Show Duplicates as necessary to review the actual data.
4. Determine which results are valid, and delete the results that are not.
5. Export CWM XML

**Define Unified Schema PF Keys**

Before defining Unified Schema PF Keys, see [How to Use Unified Schema PF Keys](#).

**Note:** If a CWM file containing PF Keys was imported as a target table, those relationships are listed on this diagram. If no additional Unified Schema PF Keys are required in the project, you do not need to perform this procedure and can refresh Unified Schema PF Key Statistics now.
Define a Unified Schema PF Key

About this task

To define Unified Schema PF Keys:

Procedure

1. Display the Unified Schema PF Keys Window. Existing Unified Schema PF Keys, if any, are listed in the Connected Tables list and in the diagram. Target tables that are not related to any other target tables are listed in the Unconnected Tables list.

2. If necessary, display the appropriate target tables as described in Unified Schema PF Keys Diagram.

3. Define a Unified Schema PF Key. There are two methods.
   - If a Unified Schema PF Key already exists between the two tables, select it in the diagram and then click the + icon in the Unified Schema PF Keys grid. A new expression containing the target table names is added to the grid. Continue with Step 4.
If no Unified Schema PF Key exists between the two target tables, use the following procedure to create one.

a. In the diagram area of the window, click once in the target table containing the foreign key column. The actual column name does not need to be visible.

**Note:** If there are many target tables in the diagram, click once in the target table in the diagram and then select *Selected Objects* from the *Display Mode* dropdown menu. (All non-related target tables disappear from the diagram.) Now click on the name of the target table with the primary key in the [Unified Schema PF Keys List] and drag it into the diagram, and continue with Step b.

b. Click and hold the mouse in the white area of the target table with the foreign key, then drag to the white area of the target table with the primary key.
c. Release the mouse in the white area of the target table with the primary key. The PF Key is added to both target tables in the Connected Tables list, and an entry is added in the Unified Schema PF Keys Grid. A new solid line connects the target tables in the diagram, with the arrow pointing to the target table with the primary key.

4. Click the new connecting line to select it, then click the dropdown arrow in the Unified Schema PF Key SQL expression field to display the The Expression Editor.
5. Complete the SQL expression by specifying the correct primary and foreign key columns. The format of the Unified Schema PF Key expression is:

**Results**

\(<\text{Predicate 1}> \text{ AND } \text{<Predicate 2}> \text{ AND } ... \text{ AND } \text{<Predicate } n>\>

- Self joins are not allowed in a manually-defined Unified Schema PF Key; however, self joins are allowed in PF Keys imported in a CWM file.
- Composite keys are supported.
- Predicates are \(<\text{Left Expression}> \text{ <operation> } \text{<Right Expression>}>\), where ‘left expression’ and ‘right expression’ refer to the target columns, functions or constants.
- Allowed operations are \(=, \neq, \leq, \geq, \text{ LIKE, NOT LIKE}\)
- There must be at least one target column reference on each side, meaning a comparison to a constant is not allowed. For example, \(P.C1=5\) is not a valid PF Key.
- NULL is not allowed. For example, \(A \text{ IS NULL}\) is not a valid predicate.
- Do not mix references to the primary or foreign target column in the same predicate. For example, \(P.C1 + F.C3 = F.P4\) is not valid.
The Unified Schema PF Key is defined. You can now refresh Unified Schema PF Key Statistics.

**Edit a Unified Schema PF Key**

**About this task**

Edit a Unified Schema PF Key in the Unified Schema PF Keys Grid by clicking in its Expression field. The Expression Editor opens, allowing you to modify the expression.

*Note:* Always refresh Statistics after editing a Unified Schema PF Key.

**Refresh Unified Schema PF Key Statistics**

**About this task**

The Re-run Step button allows you to refresh the Unified Schema PF Keys statistics. Before performing this action, see How to Use Unified Schema PF Keys.

**Procedure**

2. Verify that the task Options are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   **Note:** Modifying task options is an advanced activity. Options are described in "Reference of Task Options." Modifying options and creating new option sets is described in "Option Sets."

3. Click Run to refresh the Unified Schema PF Keys.

   **Note:** Although the Processing Options dialog closes immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the task status.

### Delete a Unified Schema PF Key

#### About this task

To delete a Unified Schema PF Key, highlight the row in the Unified Schema PF Keys Grid and click the Delete icon.
Review Unified Schema PF Keys

This section describes how to review the Unified Schema PF Keys.

Discovered Statistics

Discovery computes statistics for each target column in a Unified Schema PF Key. Statistics are calculated differently for the primary and the foreign part of the expression.

- **Selectivity**: the Cardinality of the composite key’s value divided by its row count.
  
  A high Selectivity means that most of the values in the composite target expression are unique. Low selectivity means most values are duplicates.

  Valid Unified Schema PF Keys usually have high Selectivity on the primary side. Selectivity on the foreign side may be low.) When both keys have low Selectivity, the match is probably an incidental value overlap.

- **Row Hit Rate**: the number of rows in the primary target table that have corresponding values (hits) in the foreign target table, divided by the total number of rows.
  
  A high Row Hit Rate indicates a one-to-one relationship where most of the values in the key are also in the other key. A low Row Hit Rate indicates most of the values in this key don’t appear in the other key.

  In a valid Unified Schema PF Key, one or both sides usually have a high Row Hit Rate. However, a valid primary key can have a low Row Hit Rate if it also has high Selectivity. This combination means the primary key has many unique values and the foreign key contains only a few of those values.

- **Value Hit Rate**: the number of unique values in composite target expression A with corresponding values (hits) in target column B, divided by the Cardinality of target column A.
  
  A high Value Hit Rate indicates most of the unique values in the key are also in the other key. A low Value Hit Rate means most of the unique values don’t appear in the other key.

  In a valid key, one or both sides usually have a high Value Hit Rate. A valid PF Key can have a low primary key Value Hit Rate and a high foreign key Value Hit Rate; this indicates the primary key contains most of the foreign key’s unique values along with many more unique values.

These statistics help you focus on relationships that are potentially meaningful, and to assess whether a relationship is useful or valid. A ‘perfect’ Unified Schema PF Key has a primary key Selectivity of 1 with the foreign key having a Row Hit Rate and a Value Hit Rate of 100. However, remember to always check the Data...
Preview before determining whether a Unified Schema PF Key is valid. This displays the actual rows that match or do not match the expression.

**Reviewing the Statistics**

**About this task**

In the PF Keys window, examine the **Discovered Statistics**. These statistics do not change unless the source mapping changes, or unless there are changes to the data in the source columns.

The main activities to be performed in the Unified Schema PF Keys window are:

- Delete incorrect or unacceptable Unified Schema PF Keys.
- As needed, modify existing Unified Schema PF Keys or create new ones to obtain the highest hit rates.

To accomplish these goals, you may need to meet with subject matter experts or research the data outside of Discovery. You can export the unified schema as CWM XML. See **Export CWM XML**.

One way of working in this screen is described below.

**Procedure**

1. **Zoom** in and out on the diagram as needed to locate and identify all of the target tables.
2. Rearrange the **Unified Schema PF Keys Diagram** as needed to focus on the first set of related target tables you will work on.
3. Click on each Unified Schema PF Key relationship in the **Unified Schema PF Keys Diagram** or in the **Connected Tables** list to display its expression(s) in the **Unified Schema PF Keys Grid**.
4. Verify that each PF Key relationship is valid and of high quality.
   - If a relationship has more than one Unified Schema PF Key expression, compare the target column names and statistics to make sure the primary expression is the best one. Change the primary Unified Schema PF Key if necessary.
   - Use **Show Hits**, **Show Misses**, and **Show Duplicates** as necessary to examine the data.
5. As needed, define additional Unified Schema PF Keys or edit the existing ones to improve the statistics.

**The Unified Schema PF Keys Window**

**About this task**

As soon as you add a target table, it is shown in the **Unconnected Tables** list, even before you add target columns to it.

After defining Unified Schema PF Keys or importing a CWM file containing Unified Schema PF Keys, the **Connected Tables** list shows all target tables in the project with relationships to each other. The **Unified Schema PF Keys Diagram** displays a graphical representation of the Unified Schema PF Keys for each target table.
When you complete the steps in the Unified Schema tab and refresh Unified Schema PF Key statistics, the Unified Schema PF Keys Grid is populated with the statistics.

Common Window Elements describes the menus, tabs, and buttons (like Validate Step and Activity Viewer) that are common to all windows.

**Unified Schema PF Keys List**

**About this task**

The Unified Schema PF Keys list shows the target tables in the project, grouped by **Connected** or **Unconnected**. Every target table in the project is shown in this list, either in **Connected Tables** or in **Unconnected Tables**.

You can expand and collapse the lists to view the contents more easily. Doubleclick on the list name, or click the + and - icons, to expand and collapse each list.

Unified Schema PF Keys List Icons

Unified Schema PF Key relationships are indicated by an arrow (->). The arrow points to the primary target table, so the foreign target table is first in the pair and the primary target table is second.
Connected Tables

This list shows all target tables in the project that contain a column with a relationship to a column in another target table in the project. Under each target table is a list of all other target tables with a PF key relationship with this target table. Each PF Key appears twice in the Connected Tables list: once under the primary target table and again under the foreign target table.

In the example below, the first group shows that there is a foreign key in HQ_ADDRESSES referencing a primary key in HQ_EMP. The HQ_ADDRESSES->HQ_EMP relationship is repeated under the HQ_EMP target table.
Unconnected Tables
The target tables in this list do not contain any Unified Schema PF Key relationships with any other target tables in this project.

Unified Schema PF Keys Diagram

See the [Diagram Panes](#) description for general information about working in diagrams, including:

- Selecting Tables and Connections
- Columns in the Diagrams
- Primary Key icon
- Zoom
- Changing the Table Size
- Moving Tables
- Moving Connections

The Unified Schema PF Keys diagram displays all Unified Schema PF Keys, connected and unconnected target tables in the project at the same time. You can zoom out to see all objects at once or zoom in on a particular area. Selecting a target table or connection in the [Connected Tables](#) or [Unconnected Tables](#) list shifts the diagram to display that item.

You can change the kinds of items displayed by selecting a different [Display Mode](#).
In the Unified Schema PF Keys diagram, Unified Schema PF Keys are shown as solid lines with an arrow pointing to the target table containing the primary key.

**Working with a Large Diagram**

When you rearrange target table icons and relationship lines in a Unified Schema PF Keys diagram that contains many target tables and relationships, it can take a long time to load the target tables or refresh the view after rearranging the diagram. The diagram may also be too large to see at one time.

**About this task**

When working with a diagram containing many target tables, you can lessen this time by searching in the Unified Schema PF Keys list for a specific target table, selecting it in the diagram, and then changing the Display Mode to a more restrictive filter, such as **Unified Schema PF Keys Only** or **Selected Objects’ Unified Schema PF Key**.

**Display Mode**

**About this task**

Changing the display mode allows you to focus on a smaller part of a large, complex diagram.

The available display modes are listed below. An example after each mode shows how the same diagram would appear using that mode.

**Note**: Depending on the target table currently selected, the screen layout may change when you choose a different view.

- **View All**: displays all Unified Schema PF Keys in the project.
- **Selected Objects**: Displays only the target tables and Unified Schema PF Keys that are directly related to the object (target table or connection) currently selected in the diagram or in the [Unified Schema PF Keys List](#).

---

**Unified Schema PF Keys Grid**

**About this task**

When you select a Unified Schema PF Key relationship in the [Connected Tables](#) list or in the [Unified Schema PF Keys Diagram](#), information about that relationship is displayed in the Unified Schema PF Keys grid.

The Unified Schema PF Keys grid displays all Unified Schema PF Keys between two target tables. Each Unified Schema PF Key is shown as a SQL expression.
Unified Schema PF Keys Grid Tools

The following standard tools are available for viewing and manipulating the data in the Unified Schema PF Keys grid. Also see the Grid Panes description for common actions you can perform in a grid.

- Add a Unified Schema PF Key
- Delete a Unified Schema PF Key
- Refresh Statistics for the selected Unified Schema PF Key
- Column Chooser
- Unified Schema PF Keys Data Preview - Show Hits, Show Misses
- Unified Schema PF Keys Data Preview - Show Duplicates

The following common information is displayed in the Unified Schema PF Keys grid by default:

- SQL Expression of the Unified Schema PF Keys
- Row Hit Rate
- Value Hit Rate
- Selectivity
- Notes

Unified Schema PF Keys Data Preview

This section describes Unified Schema PF Keys Data Preview Show Hits, Show Misses, and Show Duplicates.

Show Hits, Show Misses

About this task

When you select a Unified Schema PF Key and click Show Hits, the Data Preview displays the hits and misses for the selected expression.

- Show Hits displays the rows in the target columns that have hits in the other target column. These are the rows in the Row Hit Rate statistic. You can Focus on individual rows.
- **Show Misses** displays the rows in the target columns that do not have hits in the other target column.

You can also [Select Preview Criteria](#).

---

**Show Duplicates**

**About this task**

**Show Duplicates** displays source, query and target data for the selected expression when the binding condition results in duplicated rows. It can be useful to examine the duplicates when either the source column or the target column **Selectivity** is less than 1.

To analyze duplicate keys, select an expression in the **Primary Duplicates** pane and click the **Focus** icon. The corresponding rows from the primary and foreign target tables are displayed in the **Primary** and **Foreign** panes.
Chapter 3. Discovering data associations
Chapter 4. Discovering data transformations

In transformation discovery projects, you discover transformations between a source data set and a target data set. The types of relationships discovered include joins, transformations, binding conditions, and data rules.

Generate Target Matches

You can generate target matches.

Overview

Target matches are value overlaps between a source column and a target column. This is the first step in which Discovery performs source-to-target comparison.

Instead of having Discovery discover target matches, you can manually define them in this screen. However, you must have very good knowledge of the data to do this.

Target Maps

Target maps are pairs of source and target data objects, and are represented as diagrams. Each target map consists of at least one source data object and at least one target data object, where one or more of the target columns match one or more of the source columns.

Target Matches

Target matches are the actual column matches within a target map. A target match consists of one target column and one source column. The relationship between the columns can be any of the following:

• one-to-one
• one-to-many
• many-to-one
• many-to-many
• data rule

The following example shows a source data set with three data objects, and a target data set with four data objects. Each data set contains a data object describing an order entity, a product entity, and a customer entity. Three target maps have been discovered: one that matches the order entity data objects, one for the product entity data objects, and one for the customer entity data objects.
Discover Target Matches

About this task

To advance to the Target Matches window without performing Target Matches processing first, click the Target Matches tab from the Data Objects window. The Target Matches window opens with an empty Target Matches list and diagram pane. This allows you to Define a Target Map instead of having Discovery discover them.

Note: For some projects, you generate target matches immediately after you Perform Column Analysis and then Define a Map. When you do this, use the Re-run Target Matches process to discover the target matches.

To perform Target Matches processing:

Procedure

1. Click Run Next Steps... in the Data Objects window. The Processing Options Dialog appears with Target Matches selected. The Target Matches step has no sub steps.
2. If desired, drag down the **Steps** arrow to execute additional steps at the same time.

3. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   **Note:** Modifying task options is an advanced activity. Options are described in [Reference of Task Options](#). Modifying options and creating new option sets is described in [Option Sets](#).

4. Click **Run** to execute Target Matches processing. This is performed on all tables in the data sets. If the procedure does not complete correctly, see [Troubleshooting Task Execution](#).

   **Note:** Although the Target Matches window is displayed immediately, it may take some time for the actual data processing to complete. Check the [Activity Viewer](#) for the status of the task.

### Re-run Target Matches

The **Re-run Step** button allows you to repeat the Target Matches discovery step as individual actions, while you are in the Target Matches window.

### About this task

Several behaviors are important to understand regarding re-running target matches.
If an already-discovered target map exists between a source and target table and have not made any modifications to the map, Discovery does not rediscover the target map.

If you have modified a discovered map, the map is rediscovered and the changes are lost.

Manually-created target maps are deleted.

**Note:** Regenerating target matches is likely to affect maps, because binding conditions and transformations are partly based on target matches. If you have existing maps, re-run the Maps step after performing any action that generates new target matches, including manually creating a new target match.

Use **Re-run Step** in the Target Matches window if:

- You clicked directly on the Target Matches tab to manually create target matches after data objects, instead of using **Run Next Steps** from the Data Objects window. Now you want Discovery to automatically discover target matches.
- You ran the Target Matches task previously and then created or modified some of the discovered target matches. Now you want Discovery to rediscover target matches based on your changes.
- You changed the data sets in any way and need to repeat Target Matches discovery to update the results.

**Procedure**

1. Click **Re-Run Step** in the Target Matches window. The **Processing Options** dialog appears. The Target Matches step has no sub steps.
2. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.
Note: Modifying task options is an advanced activity. Options are described in "Reference of Task Options". Modifying options and creating new option sets is described in "Option Sets".

3. Click Run. Discovery discovers Target Matches for all data sources (database tables and text files) in the data sets.

Results

Note: Although the Target Matches window is displayed immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.

If this is the first time you are running the Target Matches step, the Target Matches List and Target Matches Diagram remain empty after the step is completed. To view the discovered target matches, click Show All.

Defining Target Maps

You can define a target map manually. However, you need to be familiar enough with the data to know which data objects map to each other.

Define a Target Map
You can define a target map.

Procedure

1. In the Target Matches list, click the + icon.

2. In the New Target Match screen, click on the source data object to use in the target map.
3. Click on the target data object to use in the target map.

4. Click **OK**.

   An entry is added in **Show All** and the tables in the data objects appear in the **Target Matches Diagram**.
   - If you have already discovered or defined relationships between the source and target tables in the new data object, the tables are automatically connected to each other, and you are finished. Click the **Refresh Statistics** icon to generate statistics for the new target match. You can also **Define a**
Target Match for the new target map.

- If the tables are not already related, as shown below, continue with Define a Target Match to define relationships between the source and target tables.
Delete a Target Map

Procedure

To delete a target map, select it in the Target Matches List and click the X icon. The target map is deleted, but the target matches it contains remain in the Show All list. Subsequent data discovery will still use these target matches.

Define a Target Match

Procedure

To define a target match, you need to know which tables in the source and target data objects contain related columns.

There are several ways to define a target match:

- Define a Target Match in the Target Match Diagram
- Define a Target Match in the Show All List
- Define a Target Match in the Target Matches Grid

Define a Target Match in the Target Match Diagram

Procedure

1. In the Target Matches Diagram, rearrange the data objects, if necessary, so the relevant source and target tables are both visible.
2. Click anywhere inside the relevant source (blue) table, and drag to anywhere inside the relevant target (orange) table, then release. A dashed grey line appears as you draw.

3. A target match connection is created between the two tables, and a blank entry is added in the Target Matches Grid.
4. Under **Source Column**, click on Choose a Column to display a dropdown list of all columns in the selected source table, and select a column.

5. Under **Target Column**, click on Choose a Column to display a dropdown list of all columns in the selected target table, and select a column.
6. Click the **Refresh Statistics** icon to generate statistics for the new target match.

**Results**

At this point the target match is defined and ready to use in further discovery.

**Define a Target Match in the Show All Matches List**

**About this task**

This procedure adds a target match from the Show All list.

**Procedure**

1. Click **Show All** to display the list of existing target matches.
2. In the **Show All Target Matches** popup, click the + icon to display the **Add New Match** dialog.

3. In the **Add New Match** dialog, click on the top **Source Data Set** field to display a dropdown list of all tables in the source data set. Select the desired source table.
4. In the second **Source Data Set** field, click to display a list of all columns in the selected table. Select the desired column.
5. In the two **Target Data Set** fields, select the target table and column in the same way, then click **OK** to close the **Add New Match** dialog. The new target match is added to the end of the list.

6. Select the new target match in the **Show All Target Matches** list, then click **Refresh Statistics** to discover statistics for the new target match. The statistics are displayed along with the relationship type.

### Define a Target Match in the Target Matches Grid

**About this task**

You can define a target match in the **Target Matches Grid**.

**Procedure**

1. In the **Target Matches Grid**, click the **Add** icon.

![Add icon](image1.png)

A new blank entry is added to the grid.

![Add icon result](image2.png)
2. Under **Source Column**, click on **Choose a Column** to display a dropdown list of all columns in the selected source table, and select a column.

![Source Column Diagram](image1)

3. Under **Target Column**, click on **Choose a Column** to display a dropdown list of all columns in the selected target table, and select a column.

![Target Column Diagram](image2)

4. Click the **Refresh Statistics** icon to generate statistics for the new target match.

**Results**

At this point the target match is defined and ready to use in further discovery.
Delete a Target Match

Procedure

To delete a target match, select it in the Target Matches Grid and then click the Delete icon. The target match is deleted and will not be used in subsequent processing.

Review Target Matches

The main task in the Target Matches tab is to identify target matches that are not useful or not valid, and delete them before you proceed further. Poor target matches result in unreliable binding conditions, requiring increased clean-up effort during the Maps task.

About this task

You can sometimes identify poor target matches based on the column names involved. For example, Discovery might identify a strong data match between the SHIPPER_ID and EMPLOYEE_ID columns of two tables. Columns of this type often consist of sequential integers, so numerically, there may be a very strong overlap between such columns, but logically the target match is meaningless.

As you review the target matches, delete any obvious mismatches in the Target Matches Grid or Show All. If there is any doubt about the data in a particular target match, use Target Matches Data Preview – Show Hits and Target Matches Data Preview – Show Misses to display the actual data.

- Sort the results by Row Hit Rate. Depending on the goal of the mapping project, the statistical values (source and target Row Hit Rate, Value Hit Rate, and Selectivity) may identify target matches that can be deleted.
- Depending on the goal of the project, source and target columns with column names that are obvious mismatches may be deleted.

Target Matches window

If you display this window before discovering Target Matches, the Target Matches list and the diagram are empty. If you defined target matches manually, the only target matches in the window are the ones you have already created.

Common Window Elements describes the menus, tabs, and buttons (like Validate Step and Activity Viewer) that are common to all windows.
**Target Matches List**

**About this task**

The Target Matches list does not contain separate source and target data set tabs because each target map in the list contains tables from both data sets.

After running discovery, the Target Matches list contains all of the discovered column matches grouped by target match, for each data set.

Target maps do not contain all of the tables in the data sets. Each target map contains only the tables with target matches to at least one table in the other data set. Source and target tables without target matches are not in the Target Matches list. However, those tables may still be mapped if they have join conditions or other relationships to tables in the Target Matches list.

You can expand and collapse the lists to view the contents more easily. Double click the target map name, or click the + or - icon, to expand and collapse each list.

The following screen shows a Target Matches list containing two target maps. Each target map contains two sets of source and target tables that have target matches.
**Target Map Names**

Target map names are created from the source and target data object names, in the following format:

<source_do_name>-<target_do_name>

Target maps cannot be renamed, and they do not have descriptions.

**Target Map Approval Checkboxes**

Target maps and target matches do not have approval checkboxes.

If you Define a Target Map and then Re-run Target Matches your defined target map will be deleted. If you add or delete target matches to a target map and then Re-run Target Matches your changes will be lost.

**Target Matches Diagram**

**About this task**

The Target Matches diagram displays target matches (related columns) between the source and target data sets. These are based on data objects.

The Target Matches diagram displays only the selected target map. To view the tables in a target map in the diagram, click on the target map name in the Target Matches List.
You can rearrange child tables individually in the diagram. Moving the root table of each data object moves the data object as one unit.

See the Diagram Panes description for general information about working in diagrams, including:

- Colors of Source and Target Data Sets
- Selecting Tables and Connections
- Columns in the Diagrams
- Primary Key
- Zoom
- Changing the Table Size
- Moving Tables
- Moving Connections

**Show All**

This option displays a popup window listing all discovered target matches in the project.

**About this task**

The standard tools for Grid Panes are available for viewing and manipulating the data in the Show All grid.

In addition, you can choose the sort order for columns displaying both percentage and fraction statistics. Select the desired sort method in the dropdown menu and then click in the header of a column to sort the rows according to the results in that column.

- **Sort by Rate** sorts the column by the percentage.
- **Sort by Count** sorts the column by the numerator.
The **Data Types, Mixed** column indicates whether the column contains both numeric and nonnumeric data types. When a column has mixed data types, preview the data to determine whether the results are accurate.

The following example shows the rows sorted by the column **Source Row Hit Rate**, first with **Sort By Count** and then with **Sort By Rate**.

---

**Target Matches Diagram Lines and Arrowheads**

About this task

Tables that have columns with target matches are shown as dashed lines with arrows pointing from source tables to target tables. Clicking on a dashed line displays statistics in the **Target Matches Grid**.

Solid lines represent parent-child relationships within each data set, with arrows pointing to the child tables. Solid lines do not have statistics or other information mapped to them.

**Target Matches Grid**

About this task

When you select a source-target table connection in the **Target Matches Diagram**, all discovered target matches for that table pair are displayed in the Target Matches grid. Each row represents a target match that was discovered between a column in a source table and a column in a target table.

**Target Matches Grid Tools**

The following standard tools are available for viewing and manipulating the data in the Target Matches grid. Also see the **Grid Panes** description for common actions you can perform in a grid.

- Define a Target Match in the Target Match Diagram
- Delete a Target Map
- Refresh Statistics after adding or modifying a target match
- Column Chooser
Target Matches Data Preview – Show Hits
Target Matches Data Preview – Show Misses

Note: Always click the Refresh Statistics icon after defining a target match.

Target Matches Statistics

The columns in the grid show the Type, Data Rule (if used), Row Hit Rate, Value Hit Rate, and Selectivity for each target match.

- **Type** describes the relationship cardinality of the match:
  - one-to-one
  - one-to-many
  - many-to-one
  - many-to-many

- **Data Rule**, if populated, indicates this target match is the result of a data rule. Click on the data rule name to display the Data Rules Window.

Target Matches Data Preview – Show Hits

About this task

When you select a target match in the Target Matches Diagram and click Show Hits, the Data Preview screen displays the rows in the source and target tables that match the expression.

You can also Select Preview Criteria.
Target Matches Data Preview – Show Misses

About this task

When you select a target match in the Target Matches Diagram and click Show Misses, the Data Preview screen displays the rows in the source and target tables that do not match the expression.

You can also Select Preview Criteria.

Discover Maps

This section discusses transformation maps.

Overview

A map is a collection of SQL queries that, when applied to one or more source tables, modify the source column data to match some or all of the data in a particular target table. In most projects, the main purpose of using Discovery as the mapping tool is to automatically generate these SQL queries (the map).

Map generation is a complex algorithm, using many heuristics and discovered information. Discovery searches for meaningful maps with the data sources and creates maps with a very high likelihood of representing accurate and useful relationships between source and target tables. The component SQL clauses that make up the queries in these maps will always be valid.
You can also Define a Map manually, if you know which source and target tables map to each other. Discovery will discover the bindings and transformations for the source and target tables.

Discover Maps
About this task

To skip automatic map discovery, click the Maps tab from the Target Matches tab. The Maps tab opens with no entries in the Maps list. You can now Define a Map.

To discover Maps:

Procedure
1. Click Run Next Steps in the Target Matches window. The Processing Options Dialog appears with Maps selected. The Maps step has no sub steps.

2. Verify that the task Options are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   Note: Modifying task options is an advanced activity. Options are described in "Reference of Task Options". Modifying options and creating new option sets is described in "Option Sets".

3. To perform Maps processing on only some of the target maps, click Select in the Parameters pane and select one or more target maps for processing.
Discovery will not perform discovery on the unselected target maps.

4. Click Run to execute Maps processing.

   **Note:** Although the Maps window is displayed immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.

**If No Maps Were Created**

In some situations, Discovery does not discover a map. This can happen when the discovered transformations do not meet certain criteria, and Discovery determines the resulting map would not be valid or useful.

The following situations can result in Discovery not creating a map.

<table>
<thead>
<tr>
<th>Situation</th>
<th>Possible Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery did not discover one or more transformations that exceeded the Minimum Transformation hit rate.</td>
<td>Lower the Minimum Transformation hit rate and then Discover Maps again. Define a Map and then Discover Maps again. Discovery populates the defined map with all transformations that exceed the Minimum Transformation hit rate.</td>
</tr>
<tr>
<td>The only discovered transformations that exceeded the Minimum Transformation hit rate were constant.</td>
<td>None. The data does not contain transformations that Discovery considers useful in a map.</td>
</tr>
<tr>
<td>The only discovered transformations were also used in the primary binding condition expression.</td>
<td>None. The data does not contain transformations that considered useful in a map. To view the discovered binding expression, select the completed Maps activity in the Activity Viewer and select View Trace Log.</td>
</tr>
</tbody>
</table>
Repeat Map Discovery
About this task

There is no Re-run Maps step. To have Discovery attempt to automatically discover maps again, repeat the Discover Maps step. You can also manually Define a Map.

Defining a Map
About this task

You can manually define a map instead of having Discovery discover it. This is useful when you already know which source and target tables map to each other. In this case, you create your data sets and Perform Column Analysis. You can then skip some of the remaining Discovery Studio steps as follows:

- To have Discovery discover join conditions after you define the map, run Discover PF Keys before you define the map.
- To have Discovery discover binding conditions after you define the map, run Discover Target Matches before defining the map.

Define a Map
To define a map:

Procedure

1. Perform Column Analysis, Discover PF Keys (optional), and Discover Target Matches.
2. Display The Summary Window.
3. In The Maps List, click + to create a new map.

The Create New Map dialog appears.
4. Give the map a name and, optionally, a description.
5. Select the appropriate target table in the **Target Table** dropdown menu.

6. In the **Available Source Tables** list, select one or more source tables for the map, then click **Add**. You can use CTRL-click to select several tables and then
click **Add** once to add them in one step.

7. When all tables are added, click **OK** to create the map.

8. The new map is added to **The Maps List** and shown in **The Summary Window**.
If the map contains multiple source tables and you ran Discover PF Keys, the joins are automatically added. If there are multiple tables in the map and you did not discover PF Keys, Define a Join now for the tables.

Results

Continue with Generate Binding Conditions to discover the binding conditions for the map, and then Discover Transformations. Alternatively, you can manually define binding conditions, transformations, and any other necessary map expressions.

Edit a Map

You can edit the join, binding, transformation, or other expressions in a map at any time. After doing so, refresh all Maps statistics.

Add a Source Table

About this task

You can manually add a source table to a discovered or defined map in the Map Summary diagram.

Note: Only do this if you are very familiar with the data. Re-run all subsequent Maps discovery tasks after adding a source table.

• If you add a source table in The Summary Window and the source table has PF Key relationship to one of the existing source tables in the map, a join is automatically and immediately added in the Map Information Pane and in The Join Condition Grid.

• If you add a source table and Discovery does not have a PF Key relationship for it, Define a Join for the table in the Joins tab.

Procedure

1. Click anywhere in the diagram field or on a map and right-click, then select Add Source Table from the dropdown menu.
2. In the **Add Source Table** dialog, select a source table, then click **OK**. All source tables in the source data set are shown.

![Add Source Table dialog](image)

The selected source table is added to the Maps Summary diagram, but it is not connected to the other source tables or to the target table.

![Maps Summary diagram](image)

3. If necessary, **Define a Join** to another source table in the map. After a valid join is entered in the Joins sub-tab, the connection appears in the diagram.


5. **Add a binding** for this table to the primary binding condition, or **Define a Binding Condition** to the target table and make it the primary. After a valid binding is entered in the Bindings sub-tab, the connection appears in the diagram.

6. If necessary, **Define a Where Clause**, **Discover Transformations**, or **Define a Pivot Column or Source Column**. Alternatively, you can manually define binding conditions, transformations, and any other necessary map expressions.
Delete a Source Table
Delete a source table by right-clicking on it in the Maps Summary diagram and selecting Delete from the dropdown menu.

About this task
The source table is deleted from the staging data source and it will not be used in further processing.

Note: You cannot delete the target table or the last source table in a map. Instead, Delete a Map

Review Maps Summary
About this task
After you discover or define a map, use the Map Steps to refine your discovery by changing thresholds, manually searching for joins, generating data rules, and so on. You can immediately view the results of your changes by refreshing statistics and previewing data.

Review the source and target table information presented in this screen for each map.

The Maps Summary page does not display statistics or alternative expressions, so you should Review Joins, Review Binding Conditions, and Review Transformations (including any Group By expressions). Also Review Where Clauses if any were generated.

The IBM InfoSphere Discovery Quick Reference is very useful for determining how to proceed at this point, particularly the Review Defined Map Workflow and Review Defined Map Workflow. (In Discovery Studio, select the Help>Online Documentation menu option.)

When to Run Next Steps
After reviewing the information in The Summary Window run any necessary Maps discovery or refinement tasks.

For example, you may need to refine the binding condition, or improve statistics for several transformations, or generate reverse pivots. The IBM InfoSphere Discovery Quick Reference is very useful for determining how to proceed. (In Discovery Studio, select the Help>Online Documentation menu option.)

After you complete the Maps tasks, and all statistics are acceptable, you are finished with the map and can Generate Reports or Export Data.

Run Next Steps
Run the necessary Maps steps, as determined in Review Maps Summary and When to Run Next Steps. If you do not need to perform a mapping step, such as Generate Joins or Re-run Binding Conditions skip it. However, work left-to-right in the Map Steps. For example, create any necessary Where clauses before starting to refine the transformations.
Approve a Map

About this task

When you have finished all steps in a map and have determined it is complete and accurate, you can mark the map as approved.

Once a map is approved, tasks cannot be scheduled for it: the Run Next steps and Re-run Step buttons in all Maps screens become inactive. This protects the transformations and other expressions from changing. You can still manually add and delete expressions, update expression statistics, and modify notes.

Note: After a map is approved, Discovery Studio still allows you to modify results in previous screens (PF Keys, Column Analysis, etc.) as well as modify the contents of the data sets, logical tables, and sample sets. Changing any of these items can affect the accuracy of the approved map.

To run or re-run Maps steps in an approved map, uncheck the Approved box.

The Summary Window

The Summary Window presents a high-level summary of the most important information about the currently selected map. The information is taken from the remaining Maps sub-tabs.

As you work through the Maps sub-tabs to refine the map, the information shown here will change to reflect the new discoveries.

Common Window Elements describes the menus, tabs, and buttons (like Activity Viewer) that are common to all windows. Common Elements in the Maps Sub-tabs describes items common to all Maps windows.
Map Information Pane

The Map Information pane contains a summary of important information about the selected map.
Field

The following fields are in the Map Information pane:

Name:

Name of the selected map. You can edit this name here.

Notes:

An automatically-generated description if the map was discovered. You can edit the contents here, or enter notes for defined maps.

Show Map SQL:

Displays a popup containing the full set of SQL statements that define the map. You can copy the contents and paste them into another application. However, this statement cannot be used as input for an ETL application; it is only for reference.
The Query pane contains the most important information about the selected map.

The map information is taken from the remaining Maps sub-tabs.

- **Join Condition** displays the primary join condition, if any, between two or more source tables in the map. Discovery discovers this information from PF Keys, or you can [Define a Join]. If the map contains only one source table, or the source tables in the map do not use a join, <None> is shown. (Target tables do not use joins in Discovery.)
- **Binding Condition** displays the primary binding condition between the source table(s) and the target table. Binding conditions are discovered based on target matches and other information.
- **Group By** is displayed only if the map contains any many-to-one binding conditions with aggregations.
- **Where Clause** displays the primary WHERE or CASE clause, if any, between the source table(s) and the target table. If the map does not contain any where clauses, <None> is shown.
- **Transformation** displays each target column in the map that has a transformation, and the primary transformation for each target column. Discovery discovers transformations based on target matches and other information. You can collapse and expand this list by clicking on the + box.

Reverse pivots are not shown in this screen.

**Maps Summary Diagram**

The Maps Summary diagram displays the selected map. A map always has one target table (shown in orange) and can have one or more source tables (shown in blue).
You can zoom in and out on the Map Summary diagram. Also, hovering the mouse over a join or binding condition displays the SQL expression for that connection.

Discover Joins

This section discusses join conditions.

Overview

Joins are SQL statements that relate source tables to each other in a map. They are based on PF Keys, and are discovered during Discover Maps. If a project does not contain any PF keys, the Discover Maps task will not discover joins for maps in that project. The Discover Joins task allows you to discover joins across source and target tables in a map.
Not all maps use join conditions, so this tab may be empty for a selected map.

**Types of Joins Discovered**
Discovery creates inner joins. Left outer joins are not discovered, but are supported only if they are manually defined. Right outer joins are not supported.

**When to Run or Re-run Joins**

*Note: This step can generate either joins or bindings, or both. To only generate bindings, see Generate Binding Conditions.*

If you ran the previous discovery steps and discovered maps, you do not need to run or re-run the Joins step. The source tables in the map and the joins between them are based on PF Keys that have already been verified in previous discovery steps.

Run the Joins step if you Define a Map and do not know the joins between tables. The Joins step compares the column names in each source table to the other source tables in the data set. If it discovers any column names that exactly match (a natural join), it creates a join condition for those columns.

**Generate Joins**
You can discover natural joins between source tables.
Procedure

1. Click Run Next Steps... in The Summary Window. The Processing Options Dialog appears with Bindings selected.

2. To perform Joins discovery on only some of the source maps, click Select in the Parameters pane. You can now select individual source columns to be processed. Discovery will not perform discovery on the unselected columns.

3. Select the appropriate tasks for execution:
   - To discover joins only, drag the slider up to Joins and check the Joins box.
   - To discover joins and bindings, leave the slider pointing to Bindings and check the Joins box.
• To discover bindings only, leave the slider pointing to **Bindings** and do not check **Joins**.

4. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   **Note:** Modifying task options is an advanced activity. Options are described in [Reference of Task Options](#). Modifying options and creating new option sets is described in [Option Sets](#).

5. Click **Run** to execute Joins discovery processing.

   **Note:** Although the Joins window is displayed immediately, it may take some time for the actual data processing to complete. Check the [Activity Viewer](#) for the status of the task.

**Re-run Joins**

**About this task**

**Re-Run Step** allows you to repeat the Joins discovery step. In this step, you can choose to only refresh the Joins statistics instead of repeating the entire Joins discovery action.
**Procedure**

1. Click **Re-Run Step...** in the Joins window. The **Processing Options Dialog** appears with the additional option of **Only Refresh Statistics**. Click this option to refresh all of the statistics in the Joins tab, or leave it unchecked to rediscover joins.

2. To perform discovery on only some of the source maps, click **Select** in the **Parameters** pane. You can now select individual source columns to be processed. Discovery will not perform discovery on the unselected columns.

3. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   **Note:** Modifying task options is an advanced activity. Options are described in **Reference of Task Options**. Modifying options and creating new option sets is described in **Option Sets**.

4. Click **Run** to re-run the Joins step. Processing is performed on all selected source and target tables in the map.

   **Note:** Although the Processing Options window closes immediately, it may take some time for the actual data processing to complete. Check the **Activity Viewer** for the status of the task.

**Define a Join**

There are several ways to define a join condition.
Define a join when you **Add a Source Table in The Summary Window** and Discovery does not already have a PF Key for it. This can happen when you **Define a Map** instead of having Discovery discover it, or if you skipped the **Discover PF Keys** step.

**Define a join in the Join Condition grid**

**Define a Join in the Join Matches Window**

**Define an Inner Join by Dragging and Dropping Columns**

**Define a join in the Join Condition grid**

You can define a join using the + icon in the Join Condition grid.

**Procedure**

1. Click the + icon in **The Join Condition Grid**. A new entry is created, and the cursor moves to the Expression Editor.
2. Type the new expression in **The Expression Editor**.
3. Click outside of the Expression Editor when you are finished.
4. Click **Refresh Statistics**.

   **Note:** If any join condition in the list is invalid, the Refresh Statistics action will fail. Correct the invalid join condition and refresh the statistics again.

5. **Review Joins**

**Define a Join in the Join Matches Window**

You can define a join in the Join Matches window.

**Procedure**

1. If you are adding a join match to an existing join condition, select the join condition in **The Join Condition Grid**.
2. In **The Join Condition Grid** click **Join Matches** to display the Join Matches window. Each join match represents a column relationship between the source tables in the selected map.
3. Click the appropriate join match to select it.
4. Use the join match as follows:
   - Click **Add to Join Condition**. It is added to the join as an AND.
   - Click **Create a New Join Condition**. A new join condition is created.
5. If needed, edit the new join condition in **The Expression Editor**.
6. Click the **Refresh Statistics** icon.

   **Note:** If any join condition in the list is invalid, the Refresh Statistics action will fail. Correct the invalid join condition and refresh the statistics again.

**Define an Inner Join by Dragging and Dropping Columns**

You can define a symmetrical (inner) join by dragging and dropping two columns joined by PF Keys from the Join Columns grid into the Expression Editor.

**About this task**

If you drag and drop columns that are not joined by a PF Key, only the first selected column is dropped into the Expression Editor.
Procedure

1. In the joins Columns Grid select two columns that are joined by a PF Key. Ctrl-click to select the second column.

2. Drag and drop the columns from the Joins Columns Grid to The Expression Editor. Discovery adds the syntax to create a symmetrical inner join.

3. As needed, edit the syntax in the Expression Editor.

Edit a Join

About this task

Click on a join in The Join Condition Grid and edit its SQL in The Expression Editor. Or, use the Join Matches window to add a PF Key or column match to an existing join condition.

Note: Discovery creates inner joins. Left outer joins are not discovered, but are supported only if they are manually defined. Right outer joins are not supported.

Review Joins

About this task

Use Joins Data Preview: Show Hits and Misses to identify rows that do and do not match the join condition.

- Auto Filter Row to display all of the target rows for specific rows.
- Use Focus to filter the rows in the other table to display only those that are mapped to by the selected row.

Also check the Joined Rows and Joined Bound Rows in the Joins Grid Statistics

If the join condition is not acceptable, modify it or create another one until the join condition is acceptable.

Note: If you Define a Join or Edit a Join Refresh Statistics.

The Joins Tab

About this task

If the map uses joins, click on the Joins tab to view the join details. You can also use the tools on this tab to define and test join conditions manually. Remember that all join conditions refer to the source-side tables only, and do not involve the target.
The Join Condition Grid

The Join Condition grid shows all discovered join conditions for this map, with the primary join condition marked with a green dot.

About this task

Click on a join condition to display or edit the full SQL expression in the Expression Editor.

Note: If a join condition is displayed in italics, the statistics are stale. Refresh Statistics for that expression.

Primary Join Condition

When more than one join condition is listed, Discovery automatically selects the one it considers best as the primary join. The primary join condition is marked with a green dot. You can select a different primary join by clicking the radio button on another join.

About this task

After selecting a new primary join, run Only Refresh Statistics in all subsequent Maps tabs.
Joins Grid Tools

The following tools are available for viewing and manipulating the data in the Joins grid. Also see the Grid Panes description for common actions you can perform in a grid.

- Joins Approval Checkboxes
- Define a Join
- Delete a join
- Refresh Statistics after adding or modifying a join
- Joins Data Preview: Show Hits and Misses
- Join Matches
- Column Chooser

Joins Approval Checkboxes

About this task

Check the Approved checkbox to approve the primary join condition. If you select a different join condition as primary, or if you modify the primary join condition, this box becomes unchecked.

The Approval status of the primary join has no effect on later data discovery.

Joins Grid Statistics

The columns in the grid show the following information about each join in the selected map:

- **Join Condition** is the SQL expression of the join.
- **Joined Rows** is the total number of rows in all source tables that meet the **Primary Join Condition**.
- **Joined Bound Rows** is the total number of rows in all source tables that meet the **Primary Join Condition** and also meet the **Primary Binding Condition**.

Note: If any join condition in the list is invalid, the Refresh Statistics action will fail. Correct the invalid join condition and refresh the statistics again.

Joins Columns Grid

Statistics about the columns in the source tables are shown in the Columns pane on the right. Some of the statistics were discovered during Column Analysis, while others were calculated during the Discover Maps step.
The following statistics are shown in the Joins Source Columns list by default:

- Data Type
- Map Selectivity
- Load Criteria Map Cardinality
- Sparse

**Column Chooser**

You can add the following statistics using the Column Chooser:

- Selectivity
- Cardinality
- Min
- Max
- Mode
- Mode%
- Null Count
Join Matches

The Join Matches window lists all column matches, including PF Keys, between the source tables in the selected map. This data is taken from the Show All Column Matches option in the PF Keys Diagram. These column matches can be used as building blocks if you Define a Join or Edit a Join.

Joins Data Preview: Show Hits and Misses

About this task

Use this feature to review the effectiveness of the selected join(s).

When you select a join condition in The Join Condition Grid and click Show Hits or Show Misses, the Data Preview screen displays the rows in the related source tables that match or do not match the selected expression.

A separate pane appears on the left for each source table in the selected map. You can drag and drop columns, or entire tables, from the Source Columns list on the right into the appropriate source table pane on the left. Once the columns are in the source table panes, you can perform Advanced Searches, Auto Filter Row, change the sort order, and Focus.

You can edit the expression and click Refresh Data to update the statistics to determine the effects of the change, and also click Refresh Row Count to update the Joined Rows and Joined Bound Rows count. To save the change to the expression, click Apply, or click Save As New to save it as a new binding condition.

You can also Select Preview Criteria.
Discover Binding Conditions

This section discusses binding conditions.

Overview

Binding conditions relate source tables to target tables. Binding conditions can be based on target matches, and are discovered during Discover Maps. During Discover Maps, if Discovery does not discover any binding conditions, it attempts to discover data rules that can be used in Binding Conditions or Transformations.

In order for Discovery to perform further discovery on the map, each map needs at least one binding condition.
Bindings in Highly Normalized Schemas

Highly normalized schemas are characterized by extensive use of foreign keys and generated keys. In highly normalized schemas, we often see tables that contain a large number foreign keys referencing generated keys, and not many natural values. Good binding conditions are typically based on natural keys, so generated keys do not make good binding conditions.

The rest of this section uses the following figure as an example.
In this example, both the source and the target schemas are highly normalized. All IDs are generated keys and cannot be used for binding. For example, we know that ENROLL maps to REG, although there are no natural keys that allow these two tables to bind. The ideal binding condition for these columns is to bind by both course ID and student ID, but this cannot be done because both columns are generated keys. Tables PRRQ and PREREQ also contain only generated keys. Only two tables can bind naturally: STUDENT and STDNT, using the natural keys S_name. The tables COURSE and C_INFO (joined by PF Key) bind to CRS, by the natural key of the course name.

The steps to successfully map a highly normalized schema is:
1. Find tables that bind naturally.
2. **Generate Data Rules** from these bound tables.
3. Use data rules to bind tables that cannot be bound naturally.

In the above example, we generated two data rules (one for course IDs and one for student IDs), and we can use these data rules to successfully map ENROLL to REG.

Hierarchies also happen often in highly normalized schemas. In the above example, the pre-requisite hierarchy is mapped using data rules, as illustrated in map M2.

**Generate Binding Conditions**
You can discover binding conditions between source and target tables.
Procedure

1. Click Run Next Steps in the Summary Window. The Processing Options Dialog appears with Bindings selected.

2. Verify that the task Options are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   Note: Modifying task options is an advanced activity. Options are described in Reference of Task Options. Modifying options and creating new option sets is described in Option Sets.

3. To perform Binding Conditions discovery on only some of the target maps, click Select in the Parameters pane. You can now select individual source and target columns to be processed. Discovery will not perform discovery on the unselected columns.

4. If desired, drag down the Steps arrow to execute additional steps at the same time.

5. Click Run to execute Binding Conditions processing. This is performed on all selected objects in the data sets.

   Note: Although the Binding Conditions window is displayed immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.
Re-run Binding Conditions

About this task

Re-Run Binding Conditions allows you to repeat the Binding Conditions discovery step. In this step, you can choose to only refresh the Binding Conditions statistics instead of repeating the entire Binding Conditions discovery action.

Re-run binding conditions if you change the Primary Join Condition or if you perform any other actions that change the data in the source or target tables, such as Reload a Logical Table, Update Text Files, change a sample set, and so on.

Re-run Binding Conditions is performed on all tables in the map; you cannot select only certain columns for processing.

Procedure

1. Click **Re-Run Step**... in the Binding Conditions window. The **Processing Options** dialog appears with the additional option of **Only Refresh Statistics**. Click this option to refresh all of the statistics in the Binding Conditions tab, or leave it unchecked to rediscover binding conditions.

2. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   **Note:** Modifying task options is an advanced activity. Options are described in "Reference of Task Options". Modifying options and creating new option sets is described in "Option Sets".
3. Click Run to re-run the Binding Conditions step.

   Note: Although the Processing Options window closes immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.

If Binding Condition Discovery Fails
About this task

When a map contains more than one source table but no join condition is found, the binding condition discovery task fails. Define a Binding Condition and then Re-run Binding Conditions.

Defining a Binding Condition

Any existing data rules (lookup tables) between any source and target columns can be used in a binding condition.

You need to define a binding condition when:
- You have a source and target table pair that Discovery does not have a target match for.
- You Define a Map in The Summary Window and Discovery does not already have target matches for it. This can happen if you skipped the Discover Target Matches step.
- None of the discovered binding conditions are acceptable.

   Note: Discovery supports OR in binding conditions, but it does not support OR in PF Key expressions.

Define a Binding Condition

There are several ways to define a binding condition:

About this task
- Use the Target Matches button to define a binding condition.
- Display the Show All list and browse for target matches to use to define a binding condition.
- Use the Binding Condition grid:
  1. Click the + icon in The Binding Condition Grid. A new entry is created, and the cursor moves to the Expression Editor.
  2. In the Binding Conditions Columns Grid, drag a column from a source or target table to The Expression Editor. Or, type the column name into the Expression Editor.
  3. Add the appropriate operator, then type or drag additional columns from the Columns pane to the Expression Editor, editing the expression as you work.
  4. Click outside of the Expression Editor when you are finished.
  5. Click Refresh Statistics
  6. Review Binding Conditions then Only Refresh Statistics in subsequent tabs.
Edit a Binding Condition

About this task

Click on a binding condition in [The Binding Condition Grid] and edit its SQL in [The Expression Editor]. Or, use the [Target Matches] window to add a target match to an existing binding condition.

Note: See the restrictions on binding conditions in [Defining a Binding Condition].

To edit a binding condition in the Expression Editor:

Procedure

1. Click in the binding condition expression in [The Binding Condition Grid]. Its expression is displayed in the Expression Editor.
2. Edit the binding condition using [The Expression Editor]. Click outside of the Expression Editor when you are finished.
3. Click [Refresh Statistics].

Results

[Review Binding Conditions], then [Only Refresh Statistics] in subsequent tabs if needed.

Discover Aggregations (Group By)

About this task

If a binding condition is many-to-one, the data set may contain aggregations. You can discover or define the aggregation column and then discover transformations containing aggregation expressions.

Note: If you define the aggregation column, Discovery will discover transformation expressions for the target column containing aggregate functions (sum, avg, min, max, count) and constants.

Procedure

1. Expand a binding condition that is many-to-one.

2. Identify a source column in one of the following ways:
   - Click the Generate button to discover a source column that may be the aggregation column.
   - Click in the Group By field and manually enter the name of a source column in [The Expression Editor].
3. **Discover Transformations** Discovery discovers transformations for this binding condition containing aggregation expressions.

**Review Binding Conditions**

**About this task**

After you **Review Joins** (if the map contains any), use the tools in this window to verify the quality of the **Primary Binding Condition**:

- Check the binding condition **Type**. For example, many-to-many binding conditions are not likely to be valid.
- Check the SQL expression of the binding condition.

Check the **Binding Condition Grid Statistics** to determine whether they meet your criteria.

Use **Binding Conditions Data Preview – Show Data, Show Hits, Show Misses** to identify rows that do not match the binding condition.

- **Auto Filter Row** to display all of the target rows for specific rows.
- **Use Focus** to filter the rows on the other side to display only those that are mapped to by the selected row.

**Approve a Binding Condition**

**About this task**

Check the **Approved** checkbox to approve the primary binding condition. If you select a different binding condition as primary, or if you modify the primary binding condition, this box becomes unchecked.

**The Bindings Tab**

The Binding Conditions window is shown below. This window is populated after running **Discover Maps** or after you **Define a Binding Condition**.

This tab displays not only the primary (strongest and most reliable) binding condition, but also all of the alternative binding conditions that Discovery discovered but judged less reliable.
The Binding Condition Grid

For each binding condition listed in the Binding Condition grid, Discovery Studio displays the weight and type of binding condition, along with whether the binding condition is primary. Click on a join condition to display or edit the full SQL expression in The Expression Editor.

Note: If a binding condition is displayed in *italics*, the statistics are stale. Click **Refresh Statistics** for that expression, or **Only Refresh Statistics**.

You can add the following columns using the **Column Chooser**:

- Source Hits
- Source Bound Selectivity
- Target Hits
Primary Binding Condition

About this task

Whenever Discovery discovers more than one binding condition, it selects the strongest one and designates it as primary. The primary binding condition is marked with a green dot.

You can select a different primary binding condition by clicking this radio button on another binding condition.

Note: After selecting a new primary binding condition, run Only Refresh Statistics in all subsequent Maps tabs.

Discovery applies the following criteria, in the sequence listed, to determine the strength of discovered binding conditions. The binding condition with the higher weight is selected over others with lower weights; if two are equal, Discovery moves on to the next criterion in the list.

- Correlation weight
- Target hit count
- Source hit count
- Target selectivity
- Target selectivity
- Source selectivity
- Number of predicates
- Length of binding condition expression: for this criterion, Discovery selects the binding condition with the shorter expression (the one with fewer clauses).

Results

In some cases, a primary one-to-one binding condition can be ambiguous, meaning that its bound source selectivity or bound target selectivity is less than 1.0. When this occurs, the statistics for the primary binding condition and transformations may not be correct; for example, the number of correctly or incorrectly transformed rows may be greater than 100%.

Binding Condition Grid Tools

The following tools are available for viewing and manipulating the data in the Binding Condition grid. Also see the Grid Panes description for common actions you can perform in a grid.

- Approve a Binding Condition
- Defining a Binding Condition
- Delete a binding condition
- Refresh Statistics after adding or modifying a binding condition
- Binding Conditions Data Preview – Show Data, Show Hits, Show Misses
- Export Hits and Misses
- Target Matches
- Column Chooser
Export Hits and Misses

About this task

If the tables for a selected binding condition are too large to analyze visually in Discovery Studio, you can export the hits and misses to database tables saved in your staging data source.

Procedure
1. In the Bindings tab, click a row in the Binding Condition grid to select a binding condition.
2. Select Export Hits or Export Misses.
3. In the Export to Staging Database dialog, enter the required information.
- **Table Name**: name of a new table to hold the exported records
- **Action**: select **Hits** or **Misses** to export
- **Drop Existing Table**: if a table with the same name exists in the staging data source, check this box to drop it and replace it with the new table.
- **Select Columns**: check the box for each table or column that will have hits and misses exported. By default, the columns involved in the selected binding condition are selected.

4. Click **OK**. The hits or misses for the select columns are exported to the new table in the staging data source.

**Target Matches**
The **Target Matches** button displays all target matches discovered or defined between a selected target column and all source columns in the data set (not just source columns in the binding condition or the map).
### About this task

For each target match, Discovery Studio displays the following statistics that indicate its reliability:

- **Type**
- **Source Column**
- **Target Column (Reverse Pivot)**
- **Data Rule**
- **Row Hit Rate**
- **Selectivity**

As you refine each map, you can define new binding conditions based on one or more of the target matches in this grid, or add the target matches to existing binding conditions. For example, if two target matches have high statistical values, you could combine them as a binding condition and then check the results.

**Note:** If you defined or edited a target match in the Target Matches tab but did not refresh the target match's statistics, the target match is listed in this window without statistics. As a result, when you generate binding conditions, Discovery does not discover a binding condition using that target match. You can create a new binding condition with that target match or add it to an existing target match, but the binding condition **Type** will be undefined. To discover the type, select that binding condition in the Binding Condition Grid and click **Refresh Statistics**.

### Procedure

1. If you are adding a binding condition match to an existing binding condition, select the binding condition in the Binding Condition Grid.
2. In the Binding Condition Grid, click **Target Matches** to display the Target Matches window.
3. Click in the appropriate target match row to select it.
4. Do one of the following:
   - Click **Add to Binding Condition**. It is added to the selected join as an AND.
   - Click **Create a New Binding Condition**. A new binding condition is created.
5. If needed, edit the new binding condition in the Expression Editor.
**Binding Condition Grid Statistics**
The columns in the grid show the following information about each join in the selected map:
- **Weight**
- **Type**

**Binding Conditions Columns Grid**

Statistics about the columns in the source table(s) and target table are shown in the Columns pane on the right. Some of the statistics were discovered during Column Analysis, while others were calculated during the Discover Maps step.

The following statistics are shown in the Binding Conditions Columns list by default:
- **Data Type**
- **Map Selectivity**
- **Load Criteria**
- **Map Cardinality**
- **Sparse**
Column Chooser

You can add the following statistics using the Column Chooser.

- Selectivity
- Cardinality
- Min
- Max
- Mode
- Mode%
- Null Count

Binding Conditions Data Preview – Show Data, Show Hits, Show Misses

About this task

Use this feature to review the effectiveness of the selected binding condition. These statistics include the Source Hits and the Target Hits.

When you select a binding condition in The Binding Condition Grid and click Show Hits, Show Misses, or Show Data, the Data Preview screen displays the rows in the related source and target tables that match or do not match the selected binding condition.

Drag columns from the Columns pane to the Source Pane or Target Pane to view results for additional columns.
Separate panes appear on the left for the target columns and the source columns in the selected map. You can drag and drop individual columns, or entire tables, from the Columns list on the right into the appropriate pane on the left (target columns into the Targets pane, and source columns into the Sources pane). Once the columns are in the Sources and Targets panes, you can perform advanced searches, auto filter row, change the sort order, and focus.

You can edit the expression and click Refresh Data to update the statistics to determine the effects of the change. To save the change to the expression, click Apply, or click Save As New to save it as a new binding condition.

- **Show Data** displays all rows on the source and target sides.
- **Show Hits** displays the rows in the source side that yield matching values on the target side.
- **Show Misses** displays source rows that do not yield matching values on the target side.

**Discover Where Clause**

This tab allows you to discover filters that apply to specific source rows. Discovery combines these filters into where clauses.
Overview

This tab is useful when the bound rows in a map or the source row hit rate for a binding condition in the map are below 100%.

You can also Generate Where Clauses or Define a Where Clause in this tab when the results of Review Binding Conditions or Review Transformations indicate that a filter might improve the binding condition or transformation discovery results.

Another use for where clauses is to create smaller filtered samples to reduce subsequent processing time.

Where clauses are not automatically discovered during the Discover Maps step. You only need to Generate Where Clauses or Define a Where Clause as needed.

Generate Where Clauses

To create a map that filters the bound rows using the Where clause:

Procedure

1. Click Run Next Steps... in The Bindings Tab The Processing Options Dialog appears with Where Clause selected.
2. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

**Note:** Modifying task options is an advanced activity. Options are described in [Reference of Task Options](#). Modifying options and creating new option sets is described in [Option Sets](#).

The default value of **Generate additional map with reverse where clause** is **Yes**. This creates an additional map containing the bound unfiltered rows from the source and the target (a reverse Where clause).

3. To perform Where Clause discovery on only some of the columns in the selected map, click **Select** in the **Parameters** pane. You can now select individual source and target columns to be processed. Discovery will not perform discovery on the unselected columns.

4. If desired, drag down the **Steps** arrow to execute additional steps at the same time.

**Note:** Although the Where Clause window is displayed immediately, it may take some time for the actual data processing to complete. Check the [Activity Viewer](#) for the task status.
Re-run Where Clauses

Re-Run Step allows you to repeat the Where Clause discovery step. In this step, you can choose to only refresh the Where Clause statistics instead of repeating the entire Where Clause discovery action.

About this task

![Processing Options dialog]

Procedure

1. Click **Re-Run Step...** in the Where Clause window. The **Processing Options** dialog appears with the additional option of **Only Refresh Statistics**. Click this option to refresh all of the statistics in the Where Clause tab, or leave it unchecked to rediscover where clauses.

2. To perform processing on only some of the columns in the map, click **Select** in the **Parameters** pane. You can now select individual source and target columns to be processed. Discovery will not perform discovery on the unselected columns.

3. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

**Note:** Modifying task options is an advanced activity. Options are described in Reference of Task Options. Modifying options and creating new option sets is described in Option Sets.

Chapter 4. Discovering data transformations 535
4. Click Run to re-run the Where Clause step. This is performed on all selected source and target tables in the selected map.

   **Note:** Although the Processing Options window closes immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the task status.

**Defining a Where Clause**

Define a where clause when you believe a binding condition or transformation can be improved with a filter.

**Define a Where Clause**

There are several ways to define a where clause:

**About this task**

- In **The Bindings Tab**, click **Run Next Steps**. Discovery discovers where clauses between the new source table and any existing one(s).
- Use the **Filter Matches Window** to define a where clause.

   **Note:** When defining where clauses, use only columns from source tables.

**Edit a Where Clause**

**About this task**

Click on an entry in **The Where Clause Grid** and edit its SQL in **The Expression Editor**. Or, use the **Filter Matches Window** to add an existing filter to an existing where clause.

**Review Where Clauses**

After you **Generate Where Clauses**, **Re-run Where Clauses**, or **Define a Where Clause**, check each where clause expression.

   **Note:** After you define or generate a where clause, **Only Refresh Statistics** for the primary join, binding conditions, transformations, and reverse pivots.

**The Where Clause Tab**

Display the **Where Clause tab** to view all where clauses. You can also use the tools here to **Define a Where Clause** or **Edit a Where Clause** and view the results.
The Where Clause Grid

About this task

Discovery lists all discovered filters for the selected map in the Filter Matches Window. If the Max number of filter matches to combine option is set to Yes, Discovery groups the filters into where clauses consisting of from 2 to n filters, where n is the number of filters specified in Max where clauses discovered. These where clauses are shown as Alternatives in the Where Clause grid. Each where clause applies specifically to one source column in the Where Clause Columns Grid.

The best group of filters for each source column is considered the Primary Where Clause for that column.

Click on a where clause to display or edit the full SQL expression in The Expression Editor.

Note: If where clause statistics are displayed in italics, the statistics are stale. Refresh Statistics for that expression, or Only Refresh Statistics.

Primary Where Clause:
About this task

When more than one where clause is listed, Discovery automatically selects the one it considers best as the primary where clause. The primary where clause is marked with a green dot.

You can select a different primary where clause by clicking another where clause’s radio button.

If you Re-run Where Clauses, the primary where clause is not rediscovered or changed.

Note: After selecting a new primary where clause, re-run discovery for all subsequent Maps tabs.

Where Clause Grid Tools:

The following tools are available for viewing and manipulating the data in the Where Clause grid. Also see the Grid Panes description for common actions you can perform in a grid.

- Where Clause Approval Checkboxes
- Defining a Where Clause
- Delete a where clause
- Refresh Statistics after adding or modifying a where clause
- Filter Matches Window

Where Clause Approval Checkboxes:

About this task

Check the Approved checkbox to approve the primary where clause. If you select a different where clause as primary, or if you modify the primary where clause, this box becomes unchecked.

The Approval status of the primary where clause has no effect on later data discovery.

Where Clause Grid Statistics

The columns in the grid show the following information about each where clause in the selected map.

- Where Clause
- Total Rows Passing Filter
- Bound Rows Passing Filter
Where Clause Columns Grid

Statistics about the columns in the source tables are shown in the Columns pane on the right. Some of the statistics were discovered during Column Analysis, while others were calculated during the Discover Maps step.

The following statistics are shown in the Where Clause Source Columns list by default:

- Data Type
- Map Selectivity
- Load Criteria Map Cardinality
- Sparse

Column Chooser:

You can add the following statistics using the Column Chooser.

- Selectivity
- Cardinality
- Min
- Max
Filter Matches Window

The Filter Matches window lists all discovered where clauses for binding conditions. You can also review statistics about the where clauses, and Define a Where Clause or Edit a Where Clause here.

**Binding Filters Tab:**

The Binding Filters tab displays the following information for binding condition where clauses.

- Filter
- Bound Rows Passing Filter
- Unbound Rows Passing Filter

Transformation Filters Tab:

The Transformation Filters tab displays the following information for transformation where clauses.

**About this task**

You can view information about where clauses:

- Target Column (Reverse Pivot)
- Filter
- Transformation
- Correctly Transformed Rows (Hits) Passing Filter
- Incorrectly Transformed Rows (Hits) Passing Filter

![Filter Matches](image)
Procedure
1. If you are adding a where clause to an existing where clause, select the existing where clause in The Where Clause Grid.
2. In The Where Clause Grid, click Filter Matches to display the Filter Matches window.
3. Click the appropriate tab (Binding Condition Filters or Transformation Filters).
4. Click the appropriate entry in the list to select it.
5. Use the where clause as follows:
   • Click Add to Where Clause. The selected filter is added to the existing where clause as an AND.
   • Click Create a New Where Clause. A new where clause is created.
6. If needed, edit the new where clause in The Expression Editor.
7. Click the Refresh Statistics icon.

Discover Transformations

This section discusses transformations.

Overview
This tab displays all of the transformations from source columns to target columns in the selected map, along with the data rules. You can add, modify, and delete transformations and data rules in this screen and then refresh the statistics for that expression to immediately view the results.

Note: Join Conditions, Where Clauses, and Binding Conditions are not shown here.

Transformations are discovered during Discover Maps.
Transformations Not Discovered

By default, Discovery does not discover transformations between columns containing numeric and date-time data. You can make Discovery discover these transformations by changing the Discover Transformations between date/time and numeric types? option.

Level of Decimal Precision
About this task

During transformation discovery, Discovery calculates numbers to the maximum decimal places. When the results are stored in the staging data source, the scale (number of digits to the right of the decimal) may be truncated or rounded off based on the column size, datatype, DBMS, and other factors. For example, both Oracle and DB2 automatically truncate FLOAT type target columns. The exact implementation of the truncation or rounding behavior is controlled by the DBMS.

Because of this, the level of decimal precision in the calculated results may not exactly match the level of precision in the target table. Discovery interprets these rows as misses, although the calculated values may in fact be more accurate than
the values in the target table. This is one reason it is very important to review the misses in the Transformations Grid Statistics.

In order to achieve a hit rate of 100% when the misses are due to differences in scale (truncation), you can manually edit the results in the staging data source. Alternatively, you can edit the transformation by adding a `round()` function so that the scale of the calculated numbers is equal to the scale in the target column.

**Discover Transformations**

**About this task**

If you Define a Map, you can discover the transformations.

**Procedure**

1. Click Run Next Steps in The Where Clause Tab. The Processing Options Dialog appears with Transformations selected.

2. Verify that the task Options are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

   **Note:** Modifying task options is an advanced activity. Options are described in Reference of Task Options. Modifying options and creating new option sets is described in Option Sets.
3. To perform Transformations discovery on only some of the source and target columns, click Select in the Parameters pane. You can now select individual source and target columns to be processed. Discovery will not perform discovery on the unselected columns.

4. If desired, drag down the Steps arrow to execute Discover Pivot Columns at the same time.

5. Click Run to execute Transformations processing.

**Note:** Although the Transformations window is displayed immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.

### Re-run Transformations

#### About this task

Re-run Transformations allows you to repeat some or all of the individual Transformation discovery actions: correlations, transformations, data rules, or transformation filters. Re-run transformations if you want to discover conditional transformations, discover transformations, generate a data rule from the transformations tab, or discover correlations. You can also refresh statistics.

#### Procedure

1. Click Re-Run Step in the Transformations window. The Processing Options Dialog appears with the additional option of Only Refresh Statistics. Click the appropriate option(s) to be executed.

**Note:** Discovering conditional transformations for a large number of columns can have a significant performance impact. It is recommended to perform conditional transformation discovery on selected columns only.
2. To perform processing on only some of the columns in the map, click **Select** in the **Parameters** pane. You can now select individual source and target columns to be processed. Discovery will not perform discovery on the unselected columns.

   - **Correlations**: Discovery automatically performs correlation discovery during a task’s execution, whenever it is necessary. This option allows you to force Discovery to perform correlation discovery.
   - **Transformations**: This repeats the transformation discovery executed during **Discover Maps**. Transformation discovery results depend on Correlation discovery results.
   - **Data Rules**: Discovers data rules (lookup tables).
   - **Conditional Transformations**: **Conditional transformations** are CASE statements that improve the **Primary Transformation** hit rate. Conditional transformation discovery results depend on Transformation discovery results.
   - **Only Refresh Statistics**: refreshes transformations statistics.

**Note:** When **Only Refresh Statistics** is selected, you cannot select any other options.

3. Verify that the task **Options** are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.
4. Click Run to re-run the Transformations step. This is performed on all selected source and target tables in the selected map.

   Note: Although the Processing Options window closes immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.

Define a Transformation
About this task
There are several ways to define a transformation:
• Click the Add icon, then use The Expression Editor to create the SQL expression for the new transformation. (You cannot define correlations or fuzzy matches, so the Add icon is inactive when Correlations or Fuzzy Matches are displayed.)
• Use the Filter Matches window.

   Note: After defining a new primary transformation, run Only Refresh Statistics on that transformation.

Edit Transformations
About this task
Select a transformation and then use The Expression Editor to edit the transformation.

   Note: After editing a transformation, refresh its statistics.

Review Transformations
About this task
To verify the quality and effectiveness of each primary transformation, check the Transformations Grid Statistics.

   Also use Transformations Data Preview (Show Data, Show Hits, Show Misses) to identify rows that do not match the transformation.
   • Auto Filter Row to display all of the target rows for specific rows.
   • Use Focus to show the corresponding query results and source rows.

   If a transformation is not acceptable, Edit Transformations or Define a Transformation until it is acceptable.

   Note: If statistics are stale, Refresh Statistics.

The Transformations Tab
About this task
The Transformations tab displays all primary transformations for each column in the target table. When you select a transformation in the list, the expression for the transformation appears in The Expression Editor.
During Discover Maps, if Discovery does not discover any binding conditions, it attempts to discover data rules that can be used in Binding Conditions or Transformations.

To view the alternative transformations for each column, expand a transformation in the list by clicking its + icon. All Alternatives are shown, along with statistics for each. You can also view the Correlations, Fuzzy Matches, and Filter Matches for each primary transformation by clicking on those tabs in the expanded view. However, you cannot make correlations or fuzzy matches the primary transformation.

### The Transformations Grid

**About this task**

The Transformations grid lists all columns in the target table that have a transformation from one or more source columns. For each target column, the primary transformation is displayed. Click a target column’s + icon to display all of the discovered transformations, including Alternatives, Correlations, Fuzzy Matches, and Filter Matches for each target column. The alternative marked as the primary is the expression shown in the main list.
When you select a transformation in the list, its SQL expression is displayed in the Expression Editor. The SQL expression contains only source columns, because the transformations in this list all refer to the selected target column. For equality expressions, only the name of the source table and column are displayed.

Note: Transformation statistics in italics are stale. Refresh Statistics for that expression, or Only Refresh Statistics. Stale statistics for alternatives, correlations, fuzzy matches and filter matches are refreshed by re-running Discover Correlations.

Primary Transformation
Discovery selects the strongest transformation and marks it as the primary.
Note: Correlations and Fuzzy Matches cannot be made primary transformations.

### Alternatives

**About this task**

All discovered transformations are displayed in the Alternatives view, including the transformation selected as primary. For various reasons, the other transformations are not considered as strong as the primary but are still valid transformations.

You can designate an alternative transformation as primary by selecting the radio button in front of an alternative.

The columns in the Alternatives view are:

- Alternative Transformations: discovered transformation expressions that, for various reasons, were not selected as the primary transformation
- Hits
- Misses (Transformations Tab)

### Correlations

Correlations are the degree to which values in the selected target column correlate to each source column, given the current transformation.

Correlations cannot be made primary transformations.

The columns in the Correlations view are:

- Source Column 1: the first source column in the expression
- Source Column 2: (not always present) the second source column in the relationship
- Unique
- Sparse
- Cardinality Ratio
Fuzzy Matches

Fuzzy matches describe the degree of individual character overlap between the source and target columns.

The columns in the Fuzzy Matches view are:
- Type (Correlations View)
- Source Column
- Hits

Filter Matches

Filter matches (transformation filters) represent column relationships between the source tables in the selected target column.

When discovering filter matches, Discovery’s goal is to find filter conditions that maximize the number of correctly transformed rows, while minimizing the number of incorrectly transformed rows. The ideal filter condition is one that filters out all incorrectly transformed rows.

The Filter Matches tab contains the following columns:
- Filter
- Transformation
- Correctly Transformed Rows (Hits) Passing Filter
- Incorrectly Transformed Rows (Hits) Passing Filter

Transformations Grid Tools

The following tools are available for viewing and manipulating the data in the Transformations grid. Also see the Grid Panes description for common actions you can perform in a grid.
- Transformation Approval Checkboxes
- Define a Transformation
- Delete a transformation
- Refresh Statistics after adding or modifying a transformation
- Transformations Data Preview (Show Data, Show Hits, Show Misses)
- Export Hits and Misses
- Filter Matches
- Generate a Data Rule From the Transformations Tab

Transformation Approval Checkboxes

About this task

Check the Approved checkbox to approve each primary transformation. If you select a different transformation as primary, or if you modify the primary transformation, this box becomes unchecked.

The Approval status of the primary transformation has no effect on later data discovery. However, if you Re-run Transformations Discovery will not replace an approved primary transformation, even if it discovers a better one.
### Transformations Columns Grid

Statistics about the columns in the source table(s) involved in the selected map are shown in the **Columns** grid on the right. Some of the statistics were discovered during Column Analysis, while others were calculated during the **Discover Maps** step.

<table>
<thead>
<tr>
<th>Target Column</th>
<th>Primary Transformation</th>
<th>Hits</th>
<th>Misses</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEPARTMENT_ID</td>
<td>substr(HQ_DEPARTMENTS.DEPARTMENT_ID, 1, 30)</td>
<td>33.33 % (2/6)</td>
<td>66.67 % (4/6)</td>
</tr>
<tr>
<td>DEPARTMENT_NAME</td>
<td>HQ_DEPARTMENTS.DEPARTMENT_NAME</td>
<td>100.00 % (6/6)</td>
<td>0.00 % (0/6)</td>
</tr>
<tr>
<td>STORE_NAME</td>
<td>HQ_STORES.STORE_NAME</td>
<td>100.00 % (6/6)</td>
<td>0.00 % (0/6)</td>
</tr>
<tr>
<td>STORE_STREET_ADDRESS</td>
<td>HQ_STORES.STREET_ADDRESS</td>
<td>100.00 % (6/6)</td>
<td>0.00 % (0/6)</td>
</tr>
<tr>
<td>CITY</td>
<td>HQ_STORES.CITY</td>
<td>100.00 % (6/6)</td>
<td>0.00 % (0/6)</td>
</tr>
<tr>
<td>POSTAL_CODE</td>
<td>HQ_STORES.POSTAL_CODE</td>
<td>100.00 % (6/6)</td>
<td>0.00 % (0/6)</td>
</tr>
<tr>
<td>FAX_NUMBER</td>
<td>(&quot; &quot;</td>
<td></td>
<td>HQ_STORES.AREA_CODE</td>
</tr>
<tr>
<td>MANAGER</td>
<td>token(HQ_DEPARTMENTS.MANAGER)</td>
<td>100.00 % (6/6)</td>
<td>0.00 % (0/6)</td>
</tr>
</tbody>
</table>
The following tools are available for viewing and manipulating the data in the **Columns** grid. Also see the description for common actions you can perform in a grid.

- **Show Related Columns**
- **Filter Items in a List Pane**
- **Column Chooser**

The following statistics are shown in the list by default:

- **Data Type**
- **Map Selectivity**
- **Load Criteria Map Cardinality**
- **Sparse**

You can add the following statistics using the Column Chooser:

- **Selectivity**
- **Cardinality**
- **Min**
Transformations Grid Statistics
The columns in the Transformations grid show the following information about each transformation in the selected map.

- Target Column (Reverse Pivot)
- Primary Transformation
- Hits
- Misses (Transformations Tab)

Transformations Data Preview (Show Data, Show Hits, Show Misses)

About this task

When you select a transformation expression in the Transformations Grid and click Show Hits, Show Misses, or Show Data, the Data Preview screen displays the rows in the related source and target tables that match or do not match the selected expression.

Note: In maps that use a binding condition containing aggregate functions, the Select Preview Criteria window ignores any Where clause entered for the Query. See Discover Aggregations (Group By).

Drag columns from the Columns pane to the Targets or Sources pane to view results for additional columns.
To edit the transformation, click the **Query** tab, edit the expression, and then click **Refresh Query**. Click **Refresh Statistics** to update the statistics to determine the effects of the change. The goal is to get the **Query** data to match the **Target** data as closely as possible.

The **Hits** and **Misses (Transformations Tab)** statistics indicate the effectiveness of a transformation. However, you can view the actual rows involved in the hits and misses, as well as preview the data in the source and target tables, using these screens.

Select **Show Hits**, **Show Misses**, or **Show Data**. You can switch among these views in the **Preview Criteria Screen** window by selecting the appropriate view from the dropdown list.

The **Targets** pane appears in the upper left side of the **Data Preview** window. The lower pane displays source columns when the **Sources** tab is clicked, and the results of data queries when the **Query** tab is clicked.

You can drag and drop individual columns, or entire tables, from the **Target Columns** list into the **Targets** pane and the **Query** pane. The **Sources** pane only accepts columns or tables from the **Source Columns** list.
The standard Grid Panes tools are available for viewing and manipulating the data in the Targets, Sources, and Query grids.

Export Hits and Misses

About this task

If the tables are too large to analyze visually in Discovery Studio, you can export the hits or misses. This creates a table in the staging data source containing the hits or misses.

To do this:

Procedure

1. In the Transformations tab, click a row in the grid to select a transformation.
2. Select Export Hits or Export Misses.
3. In the Export to Staging Database dialog, enter the required information.
The fields are:

- **Table Name**: name of a new table to hold the exported records
- **Action**: select Hits or Misses to export
- **Drop Existing Table**: if a table with the same name exists in the staging data source, check this box to drop it and replace it with the new table.
- **Select Columns**: check the boxes for each table or column that you want the hits or misses exported for. By default, the columns involved in the selected transformation are selected.

4. Click OK. The hits or misses for the select columns are exported to the new table in the staging data source.

---

**Generate Data Rules**

This section describes data rules.

**Overview**

A Data Rule is a special kind of transformation based on a lookup table. When Discovery discovers certain kinds of transformations that would be too complex to be expressed as a single equation (or whose equation would be extremely complex), it generates a lookup table instead. These are represented as data rules. Data Rules discovery results depend on Correlation discovery results.
There are two categories of data rules, based on how they are created:

- Discovered data rules are discovered by Discovery during three tasks, described elsewhere in this book:
  - During **Discover Maps**, Discovery attempts to discover data rules if it does not discover any binding conditions.
  - During **Re-run Transformations** if the **Data Rules** option is checked.
- Defined data rules can be made in several ways. Instructions are given in this chapter.
  - Define a Data Rule Using an Imported Lookup Table
  - Generate a Data Rule From the Transformations Tab
  - Create a Data Rule by Defining and Populating a Table

After creating a data rule, you can **Use a Data Rule as a Binding Condition**.

If needed, you can also **Export Data Rule Lookup Tables** for use in another project.

### About Shared Lookup Tables

Shared lookup tables are database tables or text files that contain lookup tables for more than one data rule. Lookup tables can be shared when they are small and it is more efficient to combine them into a single table, or for other reasons. Instructions for creating a shared lookup table are in **Details: Share Table**.

Discovery provides a **data rule data set** specifically for lookup tables. Although you can import a lookup table into any data set — source, target, repository, staging, or data rule — we recommend that you use the data rule data set. The source and target data sets contain tables that discovery is performed on, and the presence of data rule tables in those data sets can affect the discovered results.

### Shared Lookup Table Example

The following table is an example of a shared lookup table.

<table>
<thead>
<tr>
<th>S</th>
<th>T</th>
<th>Desc</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td></td>
<td>daterule1</td>
</tr>
<tr>
<td>D</td>
<td>E</td>
<td></td>
<td>daterule2</td>
</tr>
<tr>
<td>C</td>
<td>M</td>
<td></td>
<td>daterule1</td>
</tr>
<tr>
<td>F</td>
<td>G</td>
<td></td>
<td>daterule2</td>
</tr>
</tbody>
</table>

The first two columns contain the source (S) and target (T) data.

A description (**Desc**) column is present but empty.

The **Rule** column contains the data rule keys. In this example, the data rule key for the first data rule is **daterule1**, and the key for the second data rule is **daterule2**. See **About Data Rule Keys**.
About Data Rule Keys

In a shared lookup table, each data rule that shares the table has its own unique *data rule key*. This is a string that indicates which row(s) to use for that particular data rule.

Each row used in a particular data rule contains the same data rule key. The lookup table for a particular data rule is ‘built’ by using only the rows from the shared lookup table that contain the correct data rule key.

In the **Shared Lookup Table Example** there are two data rule keys: `datarule 1` and `datarule 2`. All of the rows with `datarule 1` are used to build the lookup table for one data rule, and the rows with `datarule 2` are used to build the lookup table for the other data rule.

About Default Expressions

About this task

In certain cases, your lookup table may contain several predicates whose relationship is regular enough to be expressed as a SQL expression, along with others for which that expression does not represent a useful correlation. When such expressions can be identified, Discovery calls them default expressions and lists them in the **Default Expression** field. If no such expressions are discovered, this field displays null.

The idea is to identify any rows that can be transformed in a regular way and remove them from the lookup table. This occurs when you click **Predicates: Remove Defaults** in the **Predicates Grid**.

These expressions can be used as a part of the data rule to transform whatever subset of the source data they apply to, before applying the lookup table to the remainder. Clicking **Predicates: Remove Defaults** says, in effect, “Run all values present in the lookup table first, and then use this transformation on any values that weren’t in the lookup table.”

Define a Data Rule Using an Imported Lookup Table

This procedure describes how to define a data rule using an imported lookup table. There are two parts to this procedure:

1. **Import a Lookup Table**
2. **Define a Data Rule**

If the imported lookup table is a lookup table, this procedure assumes the imported table has a **Rule** column with a unique data rule key for each lookup table that shares this table. See **About Shared Lookup Tables**.

Import a Lookup Table (Import a Data Rule)

This procedure describes how to import a lookup table so it can be used as a data rule. We recommend that you always use the Data Rule data set for lookup tables.

About this task

The data rule table or text file is imported into the project on the Data Sets page. The data rule is then made available to the project using the Data Rules window below.
Procedure
1. Import the lookup table or text file into the project’s Data Rule Data Set. Use the Import Database Tables or Import Text Files procedure.
2. In the Data Rules Window, click + to create a new data rule entry.
3. Select the NewDataRule entry and click Import. The Import Table popup appears.

The fields are described below.
- **Data Set**: the data set in this project containing the lookup table. We recommend you always import lookup tables into the Data Rule data set.
- **Table Name**: name of the lookup table, or the name of a larger table containing the lookup table columns.
- **Source Column**: The unique name of a column in the current map’s source table of the selected map that you will use in the data rule.
- **Target Column**: The column in the current map’s target table that will be derived using this data rule.
- **Description Column**: name of the column in the lookup table containing a description of the data.
- **Table is shared with other data rules:** check this box when the specified table also contains lookup tables for other data rules. See About Shared Lookup Tables. When the data rule table is shared, fill out the following fields:
  
  - **Rule Column:** the column in the specified lookup table which contains the data rule keys.
  - **Data Type** identifies the data type of the Rule Column. If you do not know the data type, we recommend that you specify varchar(50).
  - **Data Rule Key:** the data rule key for this data rule. See About Data Rule Keys.

4. Click **OK**. The Import Table popup closes.

5. Click **Refresh** in the Predicates pane. If needed, Select Preview Criteria.

### Results

The Predicates pane is populated with the contents of the imported lookup table. The project can now use the lookup table as a data rule.
Procedure

1. In [The Transformations Grid] select a target column for the data rule from the Target Column list by clicking in the appropriate row. The screenshot below indicates that the target column for the data rule will be END_DATE.

<table>
<thead>
<tr>
<th>Target Column</th>
<th>Primary Transformation</th>
<th>Hits</th>
<th>Misses</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGIN_DATE</td>
<td>HQ_EMP.HIRE_DATE</td>
<td>100.00 % (21/21)</td>
<td>0.00 % (0/21)</td>
</tr>
<tr>
<td>EMPLOYEE_LAST_NAME</td>
<td>HQ_EMP.LAST_NAME</td>
<td>100.00 % (21/21)</td>
<td>0.00 % (0/21)</td>
</tr>
<tr>
<td>SSN</td>
<td>HQ_EMP.SSN</td>
<td>100.00 % (21/21)</td>
<td>0.00 % (0/21)</td>
</tr>
<tr>
<td>END_DATE</td>
<td>HQ_EMP.TERMINATION_DATE</td>
<td>85.71 % (18/21)</td>
<td>14.29 % (3/21)</td>
</tr>
<tr>
<td>SALUTATION</td>
<td>HQ_EMP.TITLE_OF_CORTESY</td>
<td>90.48 % (19/21)</td>
<td>9.52 % (2/21)</td>
</tr>
<tr>
<td>CURRENT_STATUS</td>
<td>substr(HQ_EMP.STATUS, 1, 1)</td>
<td>85.71 % (18/21)</td>
<td>14.29 % (3/21)</td>
</tr>
</tbody>
</table>

2. Expand the selected transformation by clicking the + in front of it. The alternatives for this transformation are shown.

<table>
<thead>
<tr>
<th>Target Column</th>
<th>Primary Transformation</th>
<th>Hits</th>
<th>Misses</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGIN_DATE</td>
<td>HQ_EMP.HIRE_DATE</td>
<td>100.00 % (21/21)</td>
<td>0.00 % (0/21)</td>
</tr>
<tr>
<td>EMPLOYEE_LAST_NAME</td>
<td>HQ_EMP.LAST_NAME</td>
<td>100.00 % (21/21)</td>
<td>0.00 % (0/21)</td>
</tr>
<tr>
<td>SSN</td>
<td>HQ_EMP.SSN</td>
<td>100.00 % (21/21)</td>
<td>0.00 % (0/21)</td>
</tr>
<tr>
<td>END_DATE</td>
<td>HQ_EMP.TERMINATION_DATE</td>
<td>85.71 % (18/21)</td>
<td>14.29 % (3/21)</td>
</tr>
<tr>
<td>SALUTATION</td>
<td>HQ_EMP.TITLE_OF_CORTESY</td>
<td>90.48 % (19/21)</td>
<td>9.52 % (2/21)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alternative Transformation</th>
<th>Hits</th>
<th>Misses</th>
</tr>
</thead>
<tbody>
<tr>
<td>HQ_EMP.RETURN_DATE</td>
<td>76.19 % (16/21)</td>
<td>23.81 % (5/21)</td>
</tr>
<tr>
<td>HQ_EMP.TERMINATION_DATE</td>
<td>85.71 % (18/21)</td>
<td>14.29 % (3/21)</td>
</tr>
<tr>
<td>null</td>
<td>66.67 % (14/21)</td>
<td>33.33 % (7/21)</td>
</tr>
</tbody>
</table>

3. Create a new entry by clicking the + icon above the grid. A new, empty line is added to the list of alternatives.
4. Type the new data rule expression in The Expression Editor using the following syntax:

\[ \text{datarule}(<	ext{datarule}>, <\text{sourcetable}.<\text{sourcecolumn}>) \]

For example:

\[ \text{datarule}(\text{DR_EMPLOYEE_ID}_0, \text{HQ_EMP.EMPLOYEE_ID}) \]

5. Click [Refresh Statistics]. The statistics for the new data rule are shown. If it is better than the primary, make it the primary.

**Generate a Data Rule From the Transformations Tab**

**About this task**

When no binding condition, or only a poor binding condition, is discovered between two tables, a data rule might create or improve the mapping.

Generated data rules are based on a target column and a source expression from the Transformations screen. The source expression is one of the alternative transformations, correlations, or fuzzy matches for that target column.

The most obvious kind of generated data rule is a source correlation based on an equality transformation: “Use the raw source value for the lookup in the data rule, as it is.” However, it is also possible to use correlations based on more complex kinds of transformations, such as arithmetic.

For example if your source = 100 and target = 1000, Discovery would discover an alternative transformation, “\text{correlation(Source} \times 10),” which you could use just like any other correlation, to define a data rule. The source column for this data rule will not use the raw source value, but rather will incorporate the arithmetic transformation—in this example, it will be 1000 rather than 100.

**Procedure**

1. In The Transformations Grid, select a target column for the data rule from the Target Column list by clicking in the appropriate row. The figure below indicates that the target column for the data rule will be CURRENT_STATUS.
2. Select a source expression from one of the selected target column's Alternatives, Correlations, Fuzzy Matches, or Filter Matches. For example, an alternative transformation with a **Cardinality Ratio** close to 1 might be improved by a data rule.

To select a source expression, expand the transformation and display the appropriate tab, then click on the desired source expression. The screenshot below shows that the source expression in the data rule will be `HQ_EMP.STATUS`.

3. Click **Generate Data Rule**.
4. In the Create Data Rule Transformation dialog, rename the data rule if desired. If needed, modify the source and target column names.

If the target column or source expression you selected cannot be used to generate a data rule, an error message appears instead of the Create Data Rule Transformation dialog. Select a different item and click Generate Data Rule again.

5. Click OK. Discovery creates a data rule for the selected target column and source expression. The data rule becomes the primary.
Create a Data Rule by Defining and Populating a Table

About this task

You can define a data rule to be used by a transformation, and then define a lookup table for use with that data rule. Lookup tables for data rules must conform to a specific format, with a required Source Column and Target Column.

To create a data rule by defining and populating a lookup table:

Procedure

1. Define a Data Rule.
2. In the Details Pane, click Details: Table Redefinition. This allows you to define a lookup table for the data rule.

The fields are described below.

- **Table Name**: name of the lookup table, or the name of a larger table containing the lookup table columns.
- **Source Column**: Name of the column in the lookup table containing the source column data. You can enter a Description for this field.
- **Target Column**: Name of the column in the lookup table containing the target column data. You can enter a Description for this field.
- **Description Column**: name of the column in the lookup table containing a description of the data. You can enter a Description for this field.
- **Data Type**: the data type of the Source, Target, or Description column. If you do not know the data type, we recommend that you specify varchar(50).
- **Table is shared with other data rules**: check this box when the specified table contains or will contain lookup tables for other data rules. See About Shared Lookup Tables. Fill out the following fields:
  - **Rule Column**: the column in the lookup table which contains the data rule keys.
  - **Data Type** identifies the data type of the Rule Column. If you do not know the data type, we recommend that you specify varchar(50).
  - **Data Rule Key**: the data rule key for this data rule. See About Data Rule Keys.
3. Click OK.
4. Click Populate. The Populate popup appears.

5. In the Map dropdown menu, select the current map.
6. In the Target Column dropdown menu, select the target table defined in Step 2.
7. Click OK.
8. In the Predicates Grid, click Refresh. This adds all predicates to the lookup table that are required to derive the target column.
9. Review the contents of the lookup table in the Predicates grid.

### Use a Data Rule as a Binding Condition

You can use a defined or discovered data rule as a potential binding condition when the data rule is associated with a target match.

**Procedure**

1. In the Data Rules Window, click Target Matches. It does not matter which data rule is currently selected in the Data Rules list.

The Target Matches popup appears.
2. Click + in the **Tables** list to add a new target match entry. The *New Target Match* popup appears.
3. Click in the first **Source** field to display a list of all source tables in the current project, and select one.
4. Click in the second **Source** field to select a column from the selected source table.
5. Select a target table and column by clicking in the **Target** fields and selecting the appropriate items.
6. Click **OK** to create the data rule target match.
7. Click **Refresh Statistics** to view the results. If the target match is acceptable, click **OK** to close the popup. The data rule is added to the **Show All** screen, which allows it to be considered as a potential binding condition.

**Results**

If you do not want to use the data rule as a target match, delete it or modify it as follows:

- To delete a target match, select the target match in the diagram area or in the **Tables** list of the Target Matches popup, then click **X** to delete it. The target match is deleted from show all.
- To modify a target match, select it in the diagram or **Tables** list, then modify the source and target columns in the grid. Click **Refresh Statistics** after modifying a target match.

**Export Data Rule Lookup Tables**

**About this task**

Once you are finished with a data rule, you can export its lookup tables into a new table.
The destination database can be any database with a JDBC connection to the Discovery Studio host. Once the lookup table is exported, you can use it however you wish, including for data migration.

**Procedure**

1. Open the project. If you have been working in it, make sure all changes are saved.
2. Display the Data Rules Window.
3. Select a data rule in the Data Rules list.
4. Click Export.
5. In the Export Data Rules dialog, enter the following information:
   - a descriptive name for the table
   - a JDBC data source, selected from the JDBC data sources on the Discovery Studio host machine
   - a user name and password that are valid for the selected JDBC data source
   You can click Test Connection to verify the connection.

6. If you want to overwrite any existing table with the same name, click Drop table if already exists.

7. Click OK. The table is saved in the specified schema.

**Data Rules List**

**About this task**

Display the Data Rules list by clicking the Data Rules tab in the Columns Grid.

Double-click a data rule in the Data Rules list to display the Data Rules Window or click the Show Data Rules icon. You can Filter Items in a List Pane in this list.
Data Rules Window

About this task

The Data Rules window displays information about all of the data rules in the project (not just in the selected map). Display this window from the Data Rules List by doubleclicking a data rule or clicking the Show Data Rules icon.
Data Rules Pane

About this task

The Data Rules pane lists all data rules in the project. Select a data rule to display its information in the Details Pane.

If you delete a data rule or change its name, all expressions in the project (including expressions in other maps) that refer to the data rule are made invalid. Importing a data rule with the same name as the original will revalidate the expressions.

Details Pane

The Details pane displays information about the data rule that is selected in the Data Rules Pane. You can also modify a lookup table here.

The following fields are displayed in the Details pane:
- **Name** is the currently selected data rule name.
- **Description** is an optional field describing the lookup table.
- **Type** identifies how the lookup table was created.
- **Status** identifies whether the lookup table was created or imported.
• **Inline** uses a CASE statement instead of a lookup table. This can be faster when the data rule contains less than 10 rows.

• **Default Expression** is the lookup result in case the source lookup value is not found. See About Default Expressions.

The following tools are available in the **Details** pane:

• **Details: Table Redefinition**
• **Details: Share Table**
• **Details: Reimport**
• **Details: Target Matches**

**Details: Table Redefinition:**

This option allows you to define (create) a lookup table to be used by the data rule, or to redefine (modify) a lookup table you created earlier. See Create a Data Rule by Defining and Populating a Table.

**Details: Share Table:**

This popup allows you to define a lookup table that is combined with other (usually very small) lookup tables. A shared lookup table represents more than one data rule.

All of the data rules that share the lookup table must have the same data types in the same columns. For example, the **Source** column in the lookup table must be only one data type, and the **Target** column must be only one data type.

See About Shared Lookup Tables.

---

**Share Table**

- **Table:** DR_EMPLOYEE_ID_1
- **Columns**
  - **Source:** EMPLOYEE_ID2, **Description:** Name of the column that contains the source value
  - **Target:** EMPLOYEE_ID, **Description:** Name of the column that contains the result/target
  - **Description:** DESCRIPTION, **Description:** Name of the optional column that contains a business description
- **Table is shared with other data rules**
- **Rule Column:** DESCRIPTION, **Description:** Name of the column that contains a rule name. This column is used to identify the rule in the associated data rule.
- **Data Rule Key:**

The fields on these dialogs are described below.
• **Table Name**: name of the lookup table, or the name of a larger table containing the lookup table columns.

• **Source**: Name of the column in the lookup table containing the source column data. You can enter a **Description** for this field.

• **Target**: Name of the column in the lookup table containing the target column data. You can enter a **Description** for this field.

• **Description**: name of the column in the lookup table containing a description of the data. You can enter a **Description** for this field.

• **Rule Column**: the column in the lookup table which contains the data rule keys.

• **Data Type** identifies the data type of the Rule Column. If you do not know the data type, we recommend that you specify varchar(50).

• **Data Rule Key**: the data rule key for this data rule. See [About Data Rule Keys](#).

**Details: Reimport**

This popup allows you to import or reimport a lookup table from a data set and define it for use with a data rule. We recommend that all data rule lookup tables are imported into the data rule data set.

**Details: Target Matches**

This popup contains all target matches that can be created with the currently defined data rule. See [Use a Data Rule as a Binding Condition](#).

**Predicates Grid**

A predicate is a source or target row in the lookup table. The **Predicates** grid displays the contents of the selected lookup table as currently defined by the data rule.

![Predicates Grid](image-url)
The **Predicates** pane contains the following columns:
- **Source** identifies the column in the source table of the selected map.
- **Target** identifies the column in the target table of the selected map.
- **Description** identifies the description column.

The following tools are available for viewing and manipulating the data in the **Predicates** grid. Also see the **Grid Panes** description for common actions you can perform in a grid.

- **Predicates: Refresh**
- **Predicates: Preview Criteria Screens**
- **Predicates: Populate**
- **Predicates: Remove Defaults**
- **Predicates: Export**
- **Predicates: Export Grid Content**

**Predicates: Refresh:**

Refresh displays or refreshes the contents of the **Predicates** pane.

**Predicates: Preview Criteria Screens:**

This option displays the **Preview Criteria Screens**.

**Predicates: Populate:**

This option allows you to select a target column to create a lookup table for. See **Create a Data Rule by Defining and Populating a Table**.

**Predicates: Remove Defaults:**

This removes the default expression. See **About Default Expressions**.

**Predicates: Export:**

This option exports a data rule lookup table to any database connection you specify. See **Export Data Rule Lookup Tables**.

**Predicates: Export Grid Content:**

This option exports the contents of the grid to a file in a variety of formats: comma-delimited, tab-delimited, Excel, HTML, or XML. For more information, see **Export Rows**.

**Discover Reverse Pivots**

This section describes reverse pivots.

**Overview**

This step is only necessary when a binding condition is one-to-many and there are no other acceptable transformations.
The Reverse Pivots tab allows you to discover pivot columns and generate reverse pivot maps. When a binding condition is one-to-many, a reverse pivot relationship might exist between the source and target columns in the tables.

When the Reverse Pivots tab is first displayed, it is empty. When you Discover Pivot Columns, Discovery looks for any target columns with source columns that reverse pivot to it. A target column with this relationship is considered a pivot column, and is listed in The Target Column Grid after discovery.

If Discovery finds pivot columns, you Generate Reverse Pivot Maps. This creates a new reverse pivot map for each pivot column and adds them to The Summary Window. Each reverse pivot map has the SQL expression for its pivot column added to its primary binding condition.

Reverse pivots cannot be expressed as a single SQL statement, so Discovery creates a new set of maps for each reverse pivot.

Discover Reverse Pivots

Generate reverse pivots for all one-to-many binding conditions to discover source rows that may improve the source-target hit rate.

About this task

You cannot select specific columns to generate reverse pivots in the Re-run Steps window. Reverse Pivot discovery is always performed on all columns.
Note: If no binding conditions are one-to-many, generating reverse pivots will not improve the accuracy of the map.

Procedure
1. Discover any pivot columns in the map. Discovery checks the map and determines if any target columns have reverse pivots to source columns, and displays the target columns in The Target Column Grid.
2. Generate reverse pivot maps. Discovery creates a new map for each pivot column and adds it to The Maps List. Each new map contains all of the transformations and correlations from the original map. The primary binding condition is appended with the reverse pivot expression for that particular pivot column.

Results
If you are certain that the map contains reverse pivots, you can Discover Pivot Columns and GenerateReverse Pivot Maps as one task. However, if you are not certain, it can be more efficient to run each step as a separate task.

The instructions below are for discovering reverse pivots in two steps.

Discover Pivot Columns
Procedure
1. Click Run Next Steps... in The Transformations Tab The Processing Options Dialog appears with Reverse Pivots selected.
2. Verify that the task Options are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.

Note: Modifying task options is an advanced activity. Options are described in the Reference of Task Options. Modifying options and creating new option sets is described in the Option Sets.

3. To perform discovery on only some of the columns in the map, click Select in the Parameters pane. You can now select individual source and target columns to be processed. Discovery will not perform discovery on the unselected columns.

4. Click Run to discover pivot columns.

Note: Although the Reverse Pivots window is displayed immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.

Modifying task options is an advanced feature. The default values yield good results for most data sets. See Option Sets for more information.
Generate Reverse Pivot Maps

About this task

The Re-run Reverse Pivot task is performed after you run Discover Pivot Columns and one or more pivot columns are found and displayed in the Target Column Grid. If no pivot columns are found, there is no need to generate reverse pivot maps.

Note: Instead of running this step separately, you can generate reverse pivot maps as part of the Discover Pivot Columns task. In the Processing Options dialog, Create a New Option Set and change the Generate Reverse Pivot Maps option to True. However, if the map does not contain pivot columns, no reverse pivot maps will be discovered.

Procedure

1. Click Re-Run Step in the Reverse Pivots window. The Processing Options Dialog appears with the option Generate Map checked. Leave it checked.

   Note: When Only Refresh Statistics is selected, you cannot select any other options.

2. Verify that the task Options are set correctly. The default values are appropriate for many projects, but values can be modified to fine-tune the results.
**Note:** Modifying task options is an advanced activity. Options are described in the Reference of Task Options. Modifying options and creating new option sets is described in Option Sets.

One option you can consider is Generate additional maps for reverse pivot columns. At the default setting, False, Discovery copies the existing transformations and correlations from the original map into the new reverse pivot map. Changing this to True causes Discovery to rediscover transformations and correlations for the reverse pivot map.

3. Click Run to discover reverse pivot maps.

**Results**

**Note:** Although the Processing Options window closes immediately, it may take some time for the actual data processing to complete. Check the Activity Viewer for the status of the task.

For each pivot column, a new Reverse Pivot map is created in The Maps List. The new map contains all binding conditions and transformations from the original map, and the expression for the reverse pivot is appended to the primary binding condition.

**Define a Pivot Column or Source Column**

**About this task**

You can define a pivot column and source pivots when Discover Pivot Columns does not find any pivot columns but you believe a reverse pivot condition exists.

**Note:** Reverse pivots cannot contain any columns that are used in binding conditions.

To define a reverse pivot:

**Procedure**

1. Click the + icon in The Target Column Grid.

A new entry for the pivot column is added in the grid.

2. Click the down arrow in the new entry and select a target column that is not used in a binding condition.
3. Click the expand icon to expand the new entry.

4. Click the down arrow of the new Source Column entry, and select a source column that is not used in a binding condition.
5. With the focus still in the source column, click the + icon again to add another Source Column entry.

6. Select another source column that is not used in a binding condition.
7. As needed, add more source columns.
8. When all source columns have been added, click the Refresh Statistics icon.
9. If the Source Hits indicate that the source columns reverse pivot to the pivot column, Generate Reverse Pivot Maps.

**Edit a Reverse Pivot**

Refresh statistics after editing a reverse pivot.

**About this task**

- To add a source column to a reverse pivot, click in an existing source column for the reverse pivot and click the + icon. In the new source column entry, select a source column that is not used in a binding condition.
To delete a source column from a reverse pivot, click in the source column and then click the X icon. The source column is deleted.

To delete a pivot column with or without source columns, click in the pivot column entry and then click the X icon. The pivot column and any associated source columns are deleted.

**Review Reverse Pivots**

**About this task**

Examine the statistics in the Target Column Grid. If the sum of all Source Hits in that reverse pivot are equal to or close to the number of unmapped rows in the pivot column, the reverse pivot is successful.

If the sum is not close to the number of unmapped rows, the reverse pivot may still be useful. Use Show Hits, Show Misses, and Show Data to verify that the Source Hits are valid. Define additional source rows or reverse pivots if necessary.

**The Reverse Pivots Tab**

The Reverse Pivots tab displays discovered pivot columns and reverse pivot maps.
The main entry in the **Target Column** grid is the pivot column. The example below shows two pivot columns listed. Each pivot column has two source columns that reverse pivot to it.

![Target Column Grid](image)

**Note:** If a pivot column is displayed in italics, the statistics are stale. [Refresh Statistics](#) for that pivot column, or [Only Refresh Statistics](#).

### Target Column Grid Tools

The following tools are available for viewing and manipulating the data in the **Target Column** grid. Also see the [Grid Panes](#) description for common actions you can perform in a grid.

- Target Column Grid Approval Checkboxes
- Define a Pivot Column or Source Column
- Delete a reverse pivot
- Refresh Statistics after adding or modifying a reverse pivot
- Column Chooser

### Target Column Grid Approval Checkboxes

Source columns for a pivot column can be approved. You can use this to keep track of source columns you have verified.

The Approval status of the source columns in a reverse pivot has no effect on data discovery.

### Target Column Grid Statistics

The columns in the Target Column grid show the following information about each reverse pivot in the selected map:

- **Target Column** is the name of the pivot column.
- **Hits** applies to the pivot column only, and is the number of rows in this column that can be derived from all listed source columns.

- **Total Hits** applies to the pivot column only. It is the number of rows in this column that can be derived from all its source columns, whether they are currently included (checked) or not.

- **Source Column** is the name of a source column that reverse pivots to the pivot column.

- **Source Hits** is the number of rows in this column that will be used in deriving the selected target column.

**Reverse Pivot Columns Grid**

Statistics about the columns in the source tables are shown in the **Columns** pane on the right. Some of the statistics were discovered during Column Analysis, while others were calculated during the **Discover Maps** step.
The following statistics are shown in the Reverse Pivot Columns list by default:

- **Data Type**
- **Map Selectivity**
- **Load Criteria Map Cardinality**
- **Sparse**

**Column Chooser:**

You can add the following statistics using the **Column Chooser**.

- **Selectivity**
- **Cardinality**
- **Min**
- **Max**
- **Mode**
- **Mode%**
- **Null Count**
Chapter 5. Managing results

After running a source data discovery (SDD) or a transformation discovery (TD) project, you can export data, generate reports, and create validation jobs.

Export Data

One you have refined and verified all results, you can export data in a variety of ways.

Before exporting a project, Prepare the Project for Export.

Prepare the Project for Export

Before exporting a project, including exporting XML, generating XML, or migrating data, perform the following steps to prepare the project.

Procedure
1. Perform project-level validation on the project.
2. Check the statistics for each object in each tab. If any statistics are stale, refresh them.
3. Repopulate all created data rules.
4. If any data rules use an imported lookup table, reimport the lookup tables into the new project's data set data source and then update the data rule's lookup table definition. This is only necessary when the new project has a different staging data source than the master project.
   For example, assume the master project is named P_0, and uses a data rule based on an imported lookup table. A data analyst imports project P_0 to a new project named P_1. Because P_1 uses a different staging data source than P_0, the imported lookup table is not available. The data analyst needs to add the lookup table into the P_1 data source data set and then update the data rule to use the lookup table.
5. Save the project.

Results

The project is now ready to be exported as described in the following sections.

- Export Full Project
- Generate Optim Export
- Export Project Objects
- Export Hits or Misses
- Generate Discovery XML
- Export CWM XML

In addition, you can Migrate Data using Discovery Generated SQL.
Export Full Project

Exporting a full project creates a set of XML files containing information about data sets, tables, column matches, PF Keys, target matches, mappings, transformations, bindings, joins, data rules, and other project-related data.

These XML files are not meaningful when viewed directly, but can be imported into a new Discovery project to re-create the original project. This is useful for several purposes: importing a project into a different Discovery Server repository, creating a copy for other analysts to work on, making a backup or archive copy, and so on.

Project Information Not Included

Most of the information in a project is included when you export it, including all map results such as joins, bindings, transformations, and so on. This information is stored in the Repository Database.

The exported project does not contain information from the Staging Data Source because this information is no longer needed after maps have been discovered; you can manipulate and refine the map results in an imported project without it.

Information not saved with exported projects includes:

- data distribution information in internal profiling tables (used to create column matches)
- JDBC connection details
- project password

In addition, data rule tables are not exported from the staging data source due to their potential size.

The contents of the staging data source can be re-created after the project is imported, if needed.

Important Considerations when Exporting Projects

When you export a project, take the following into account to ensure a successful import of the project later:

- **Project passwords** are not included in the exported project. After exporting a project, anyone can view the XML files, import the project, open it, and modify it. You can Assign a Password after the project has been imported.

- **Data rule tables** are not exported with the project. If the imported project uses the original project's staging data source, and the original data rule tables are still in that staging data source, the data rules will work. However, if the data rule tables have been removed from the original staging data source, or the imported project uses a different staging data source, the data rules will fail. Make sure that data rule tables remain accessible after the project is imported, either by using a database utility to copy the data rule tables to an accessible location or by not deleting them from the original staging data source. If necessary, you can regenerate the data rule tables in the imported project by re-creating the data rules.

- If the project is imported to a different Discovery Studio machine, the ODBC connections may be different or missing. These are not needed if you import the project only to manipulate the map results. However, if you might need to reload source tables or connect to the original data sources for other reasons,
you should note the connection names, JDBC data sources, user names, and passwords for all connections, and have this information accessible when you import the project.

**Location of Exported Project**
Discovery Studio creates a new directory using the project name and saves the exported project there as XML files. For example, if you specify a directory named `Exported_Projects` when you export a project named `Sales2006`, the exported files are created in `Exported_Projects/Sales2006/<files>.xml`. After export, you can add other files to this directory, such as text files, tables, etc.

A folder can only contain one set of project files. If you export `Sales2006` to the `Exported_Projects` directory again, Discovery Studio asks if you want to overwrite the existing project in that directory. Non-project files in the directory are not overwritten.

**Export the Full Project**
To export a project:

**Procedure**
1. **Prepare the Project for Export**
2. From the main menu, select `Project>Export Full Project`.

**Note:** If the project has been changed since it was last saved, a dialog asks you to save the changes before exporting, discard the changes and export the last saved version, or cancel the export action.

3. In the `Browse for Folder` dialog, select a directory. A new folder, using the project name, will be created in this location to save the exported files. If the target directory already contains a subfolder with the project name, Discovery asks if you want to overwrite the existing project or cancel the export.

   To create a new subfolder in the target directory, click **Make New Folder**. A new, empty folder is created in the target directory. Select the new folder and rename it, if desired.
4. Click **OK**.
5. A dialog asks if you want to include column matches and target matches in the export. (For SDD projects, you are asked about column matches and overlaps.) If they are not included, they can be re-created after the project is re-imported by repeating the PF Keys and Target Matches discovery.
   - If the purpose of exporting the project is for other analysts to work on the data, you can include the matches so each analyst does not need to repeat discovery, but the total size of the export will be larger.
   - If the export is for creating an archive or backup copy, the matches are not needed.

6. The project is saved as a set of XML files in the specified folder.
A confirmation dialog appears when the import is complete and indicates how many objects were exported. These objects are maps, tables, and so on, and do not correspond to the number of XML files in the export directory.

Results

Individual files are named according to the project and data set names.

Exporting to Guardium

You can export part or all of a project to Guardium.

Procedure
1. From the menu bar, click Project > Export > Export to Guardium.
2. From the browser window, select a valid export file type, such as *.csv. It is recommended that you name the file the same as your project name.
3. Click Save.

Export Project Objects

You can export individual maps, data rules, and option sets for import into new or existing projects.

Procedure
1. Prepare the Project for Export
2. From the main menu, select Project > Export > Project Objects.
Note: If the project has been changed since it was last saved, a dialog asks you to save the changes before exporting, discard the changes and export the last saved version, or cancel the export action.

3. In the Export Objects dialog, click the checkbox by each object to be included in the export.

4. Click Browse and select a directory. A new folder, using the project name, is created in this location to save the exported objects. If the target directory
already contains a subfolder with the project name, Discovery asks you to overwrite the existing project or cancel the export.

To create a new subfolder in the target directory, click **Make New Folder**. A new, empty folder is created in the target directory. Select the new folder and rename it, if desired.

5. Click **OK**. The objects are saved as XML files in the specified folder. An additional file named `ExportCatalog.xml` is also created in the folder. A confirmation dialog appears when the export is complete and indicates how many objects were exported.

**Generate Optim Export**

These instructions are for exporting a Discovery project for use in Optim.

**About this task**

The Optim export has the following restrictions:

- Text files in the Discovery project are not exported.
- Only physical tables are exported. Logical columns without an exactly corresponding physical column are not exported. To avoid missing logical columns in the export, make sure that each physical table uses its `<**Full Set**>` logical table as the primary.
- Each data set must have only one database connection. If a data set contains tables from two or more database connections, only the tables from the first connection are included in the exported .dbm file.

**Procedure**

1. **Prepare the Project for Export**
2. From the main menu, select **Project>Export>Export for Optim**.
3. Select the export directory.
4. Click **OK**. A new directory with the project name is created in the export directory, containing the project files.

**Results**

Discovery creates one `<project_name>.ddm` file for the project. A separate `.dbm` file is created for each data set, and separate `.ldm` and `.dbm` files are created for each data object.

The exported files can be imported directly into Optim; when you do, add the `<project_name>.ddm` file as a domain reference.

**Export Hits or Misses**

You can export information on the hits and misses between specified source and target columns. This is available for transformations and for binding conditions.

**Note:** Exported hits and misses data is only available for the primary transformation or binding condition in the grid. If you want to generate a report for one of the alternatives, you must first designate it as primary using the Make Primary button.
These exports can be quite large, so they are output not as HTML or Excel files, but as physical database tables in your staging data source. You can open them and manipulate the contents with third-party database tools.

**Overwriting Existing Reports**

In most cases, you will be generating reports as new tables in the staging data source. However, if you intend to regenerate an existing report using the same table name—in effect, overwriting an existing table—select the **Drop existing table** option.

**Reports and the Data Preview Panes**

There is a connection between a hits or misses report and the data you are displaying in the data preview panes at the time you generate a report. Both types of reports provide check boxes that let you specify which columns you want to include, from all tables on both source and target sides in the map.

**Export Hits or Export Misses - Transformations**

**About this task**

You can generate transformation hits and misses reports from the Transformations tab of any map. Because transformations necessarily refer to both the source and the target, the report contains information on source and target columns, along with the query results.

**Procedure**

1. **Prepare the Project for Export**
2. In the Transformations tab, click either **Export Hits** or **Export Misses**.
3. In the **Export to Staging Database** dialog, provide a name for the report in the **Table Name** field.
   - It is helpful to give the table a name that describes the contents.
   - The name must be valid as an actual database table name. Spaces and special characters (other than underscores) are not permitted.
This window presents a list of all columns in all logical tables in the source, target, and query results. Select the columns to include in the report. Deselect any you do not want to include.

The currently displayed columns are already selected by default. If you do not have any data displayed in the data preview panes, all tables and columns are initially selected when you open the window.

4. Click **OK** to generate the report.

**Export Hits or Export Misses - Bindings**

**About this task**

Hits and misses reports for binding conditions are similar to those for transformations. A report can be created for the source columns or for the target columns, but both types of columns cannot be included in the same report.
• Source reports provide statistics on the degree to which a specified binding condition is effective at mapping the columns on the source side to some target columns.
• Target reports show how well the given binding condition generates the columns on the target side, from any source column.

Procedure
1. Prepare the Project for Export
2. In the Bindings tab, click either Export Hits or Export Misses.

3. In the Export to Staging Database dialog, provide a name for the report in the Table Name field.
   • It is helpful to give the table a name that describes the contents.
   • The name must be valid as an actual database table name. Spaces and special characters (other than underscores) are not permitted.
4. Select one or more source or target columns for the report.

   **Note:** You cannot include both source and target columns in a single report.

   The currently displayed columns are selected by default. If you do not have any data displayed in the data preview panes, all tables and columns are initially selected when you open the hits or misses report window.

5. Click **OK** to generate the report.

**Export for InfoSphere FastTrack**

These instructions are for exporting a Discovery SDD project for use in FastTrack.

**About this task**

Discovery allows you to select business terms and assigned assets in Discovery and export them into a read-only grid. All maps are shown, even ones that are incomplete or cannot be exported. You then export the data as an InfoSphere Business Glossary export XML file, which can then be imported into InfoSphere Business Glossary.
Procedure

1. Prepare the Project for Export
2. From the main menu, select **Project>Export>Export for FastTrack** This displays the Maps Report, described below.
3. After reviewing the grid, **Export the Maps Report as a .csv File**

Results

The Maps Report allows you to filter the rows and rearrange columns until the desired data is shown.

<table>
<thead>
<tr>
<th>Source Column</th>
<th>Target Column</th>
<th>Transformation Rule</th>
<th>Tr</th>
<th>Am</th>
<th>Join</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source.ISD_SOURCE.H.E.</td>
<td>Target.ISD_SOURCE.W.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source.ISD_SOURCE.H.E.</td>
<td>Target.ISD_SOURCE.W.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source.ISD_SOURCE.H.E.</td>
<td>Target.ISD_SOURCE.W.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source.ISD_SOURCE.H.E.</td>
<td>Target.ISD_SOURCE.W.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source.ISD_SOURCE.H.E.</td>
<td>Target.ISD_SOURCE.W.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source.ISD_SOURCE.H.E.</td>
<td>Target.ISD_SOURCE.W.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source.ISD_SOURCE.H.E.</td>
<td>Target.ISD_SOURCE.W.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source.ISD_SOURCE.H.E.</td>
<td>Target.ISD_SOURCE.W.E.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See the **Grid Panes** description for common actions you can perform in this grid.

The columns in the report are described below. Columns marked with * contain map-level information and are only populated for the first row in the map.

- **Source Column**: the source column used in the transformation, in the format `<datasourcename>.<schemaname>.<tablename>.<columnname>`
- **Source Business Term**: the source column **Classification (Group Names)**
- **Target Column**: the target column used in the transformation, in the format `<datasourcename>.<schemaname>.<tablename>.<columnname>`
- **Target Business Term**: the target column **Classification (Group Names)**
- **Transformation Rule**: the source-to-target transformation expression. This is blank when the transformation is a copy-over of the source column. Null and constant transformations are listed but their corresponding **Source Columns** are empty.
- **Transformation Function**: empty column for IBM InfoSphere DataStage® use
- **Annotation**: any **Notes** entered for the transformation
- **Join***: empty column for IBM InfoSphere DataStage use
- **Filter***: the primary Where clause, if any, for the map
- **Specification Description***: the map name and the primary join expression, if any
- **Status**: the **Approved** status of the transformation. If the transformation is unapproved, this is the status of the Transformation step.
Export for Metadata Server

These instructions are for exporting a Discovery project for use in Metadata Server.

About this task

Note: This export is similar to the Export for Optim feature, but includes only the built-in (Optim/Information Analyzer) classifications. User-defined classifications and Business Glossary classifications are not included unless the classification name matches an Optim/Information Analyzer data class name. The export only includes database tables, and does not include any text files in the project.

Each data set must have only one database connection. If a data set contains tables from two or more database connections, only the tables from the first connection are included in the exported .dbm file.

Procedure

1. Prepare the Project for Export
2. From the main menu, select Project>Export>Export for Metadata Server.
3. Select the export directory.
4. Click OK. A new directory with the project name is created in the export directory, containing the project files.

Results

Discovery creates one <project_name>.ddm file for the project. A separate .dbm file is created for each data set, and separate .ldm and .dbm files are created for each data object.

The exported files can be imported directly into Metadata Server; when you do, add the <project_name>.ddm file as a domain reference.

Export Business Glossary XML

Discovery allows you to export column classifications as IBM Business Glossary terms and assigned assets.

About this task

It is a good practice to consider Business Glossary as the main repository of business terms. When new classifications and assignments are made in Discovery Studio, you should export the new term-to-column assignments to Business Glossary, delete the classifications in Discovery Studio, then re-import the glossary into Discovery Studio.

The export only includes database tables, and does not include any text files in the project.
Procedure

1. In Discovery Studio, open a project and select Project>Export>Business Glossary XML. The Export Business Terms and Assignments window appears.
2. Expand the list to show all terms and assignments. All items are selected by default.
3. Uncheck (deselect) the terms or assignments that will not be exported.
4. Enter a Database Host and Database Name that will be prepended to each item.
5. Click Browse to navigate to an appropriate directory and select or enter a name for the output file.
6. Click Export. The Export Business Terms and Assignments window closes. The export file can now be imported into Business Glossary.

Generate Discovery XML

Discovery SDD and TD projects can be exported in a generic XML format, called Discovery XML, that can be read by third-party tools such as metadata repositories.

If an SDD project contains target tables or Unified Schema PF Keys, those objects are also included in the Discovery XML export. A separate XML file is created for each data set in the project, and one XML file is created for the target tables.

Contents of the Export

During Discovery XML export, separate files are created for objects in the project. The contents of the export are different for TD and SDD project exports.

SDD project exports contain a separate file for each of the following objects.

- Each data set in the project
- All target tables in the project, including:
  - source mappings
  - join conditions
  - conflict detection rules
  - conflict resolution rules
  - Unified Schema PF Keys (if used)
- Unified Column Analysis results for all target tables, including data types and statistics

TD project exports contain a separate file for each of the following objects. You can select additional objects to be exported in the Export Discovery XML dialog.

- Each map in the project, including:
  - join condition (if used)
  - primary binding condition
  - where clause (if used)
  - primary transformations
  - reverse pivots (if used)
- Source data set
- Target data set
- Source column matches (optional)
- Target column matches (optional)
• Target matches (optional)
• Each inline data rule in the project (optional)
• Non-inline data rules in the project (optional)

**Handles and References**
All references to tables, columns, inline data rules, non-inline data rules, and data rule lookup tables are mapped to xpath-like URLs, and are portable.

Table handles in expressions use the tables’ logical names, which are unique in the project. For example, an exported column is represented as CurmePC\SOURCE\HQ_ADDRESSES\EMPLOYEE_ID.

Expressions use the column name alone. For example, consider the following transformation in a TD project export:

```
substr(HQ_EMPS.EMPLOYEE_ID, 5,3)
```

The transformation would be exported with the following changes:

- The handle HQ_EMPS is given an xpath representation of VikramPC\SOURCE\HQ_EMPS.
- The handle EMPLOYEE_ID is given an xpath representation of VikramPC\SOURCE\HQ_EMPS\EMPLOYEE_ID.

Exported datarules use the datarule name as the handle. For example, a datarule DR_EMPLOYEE_ID_6 might be given an xpath representation of salilpc\staging\DR_EMPLOYEE_ID_6.

**Discovery XML XSD**
The Discovery XML XSD is installed at C:\Program Files\IBM\InfoSphere\Discovery\Discovery Server\exerosconfig\DiscoveryAbstractMappingModel.xsd.

**Export the Discovery XML**
These instructions are for exporting an SDD or TD project in the Discovery XML format.

**Procedure**
1. **Prepare the Project for Export**
2. From the main menu, select **Project>Export>Discovery XML**.
3. (TD projects only) In the *Export Discovery XML* dialog, select any additional objects to export.

The following options are available:

- **Alternative Transformations**: Check this to include all alternative transformations for each primary transformation. When unchecked, only primary transformations are included. Correlations and fuzzy matches are never included.

- **Column Matches (up to 1000)**: Check this to include the first 1000 source and target column matches returned by the database for each map. When this is unchecked, column matches are not included in the export.

- **Target Matches (up to 1000)**: Check this to include the first 1000 target matches returned by the database for each map. When this is unchecked, target matches are not included in the export.

- **Data Rule Lookup Table Values (up to 20 per table for non-inline rules)**: In each map, includes all values of inline data rule lookup tables, and up to 20
values from each non-inline data rule lookup table. When this is not checked, data rule tables are not included in the export.

4. In the Export Discovery XML dialog, select the export directory.
5. Click OK. A new folder with the project name is created in the export directory.

Export CWM XML

About this task

These instructions are for exporting a project in the CWM (Common Warehouse Metamodel) XML format. The discovered relationships exported in this format can be imported into metadata repositories, and you can then use the metadata repositories as part of a data lineage project to understand how data flows between systems and to perform impact analysis.

The data is exported in the datatypes defined in the Target Table Schema tab.

If an SDD project contains target tables or Unified Schema PF Keys, all of those objects are also included in the CWM export. A separate XML file is created for each data set in the project, and one XML file is created for the target tables.

Procedure

1. Prepare the Project for Export.
2. From the main menu, select Project>Export>CWM XML.
3. Select the export directory.
4. Click OK. A new directory with the project name is created in the export directory, containing the project files.
   For TD projects, a single XML file is created with the project name.
   For SDD projects, all target tables are exported into a single XML file named <project_name>_USBDataSet. Each data set is exported into its own XML file with the appropriate dataset name.

Results

Before importing the XML file into the metadata repository, you need to edit the file to specify the catalog or schema.
Migrate Data using Discovery Generated SQL

Before you begin

Once a map is finished, you can use the SQL expression that represents the map, along with the lookup tables, as the basis for an ETL project. All conversions, commands and updates are performed on the tables in the Discovery staging data source; the original source and target tables are not modified in any way. The actual destination can be a table, text file, multiple files, or any other format needed.

The major steps in the procedure are:

Procedure

1. Prepare the Project for Export
2. Generate the Maps Using the Full Set Sample Set for All Source Tables
3. Prepare the Map SQL for Migration
4. Migrate the Data
5. Load the Data into the Destination

Results

1. Prepare the Project for Export

Generate the Maps Using the Full Set Sample Set for All Source Tables

1. If all source tables in each map had the Full Set sample set as primary when the map was generated, skip this step and proceed to Step 3.

For each map in which all source tables did not use the Full Set sample set when the map was generated, do the following:

a. Make Full Set the primary sample set for each source table in the map.
b. Refresh all map statistics.
c. Repopulate the data rules.

Prepare the Map SQL for Migration

1. For each map, in the Maps Summary tab, click Show Map SQL to display the SQL expression for that map.

2. Copy the entire query from the popup window and paste it into a text editor.
3. In the text editor, add a **Create** and **Insert** clause to the beginning of each map SQL query. Use the following syntax, specifying a new target table name for each map.

   ```sql
   Create Table <new_target_table> as
   ```

   For example, if the query starts as shown:

   ```sql
   select ( substr(HQ_DEPARTMENTS.DEPARTMENT_NAME, 1, 1) || '1'),HQ_DEPARTMENTS.DEPARTMENT_NAME,HQ_STORES.STORE_NAME,HQ_STORES.STREET_ADDRESS,HQ_STORES.CITY,HQ_STORES.POSTAL_CODE,
   ```

   Then the edited query is as follows, assuming a new target table name of **NEWTARGET**:

   ```sql
   Create Table NEWTARGET as select ( substr(HQ_DEPARTMENTS.DEPARTMENT_NAME, 1, 1) || '1'),HQ_DEPARTMENTS.DEPARTMENT_NAME,HQ_STORES.STORE_NAME,HQ_STORES.STREET_ADDRESS,HQ_STORES.CITY,HQ_STORES.POSTAL_CODE,
   ```

4. Save and close the text file.

**Migrate the Data**

1. For each map, run the edited SQL query to load the map's data from the source tables in the staging data source to the new target table. If needed, copy the data rule table(s) separately.

**Load the Data into the Destination**

There are different methods of loading the target data, depending on the load interface of the target application. Some examples are:

- **Vendor-specific Interface**: Some target systems provide their own migration interfaces—for example, SAP's Data Migration Interface (DMI). This is the simplest case, but it requires expertise on the migration interface.
- **Functional Load Interface**: Write code to apply each target table row as a function call.
- **File Load Interface**: Dump generated target tables into files.
- **Production Table Load**: Copy data from target tables directly into production tables.

**Generate Reports**

Once you have refined and verified all results, you can generate reports to use in the next step of your project.

Discovery Studio can generate the following reports:

- **Project Report**: lists all relationships and discovered results in the project. Also allows you to export TD project maps as .csv files compatible with IBM InfoSphere FastTrack.
- **Table Report**: contains information about how the source and target tables map to one another, based on the transformations Discovery has discovered so far.
- **Column Report**: contains subsets of the analysis performed in the context of each data source map.

**Project Report**

The project report contains a full list of all relationships and results in the project. Project reports can be exported in three formats:
Generate the Project Report

About this task

Generating a report from a large project can take 30 minutes or longer, and is memory-intensive. To provide the largest amount of free memory for this task, restart Discovery Studio and then immediately generate the report before performing any other actions. You can also check the Ignore Transformation Alternatives option to reduce the memory required.

The project reports for TD and SDD projects are slightly different.

To generate a project report:

Procedure
1. Open a project.
2. From any tab, select Tools>Reports>Project. The following dialog appears.

![Report Options dialog]

3. Select the appropriate format in the dropdown list:
   - Excel, to save as an Excel spreadsheet
   - HTML, for viewing in a web browser
   - Maps (TD projects only), to reformat the map data in a grid that can be exported as .csv files compatible with IBM InfoSphere FastTrack

   Note: To generate reports in Excel, Excel 2002 (XP) or Excel 2003 must be installed on the Discovery Studio host, and must be configured to print.

4. By default, the report includes transformation alternatives for Excel and HTML reports. To exclude these, check Ignore Transformation Alternatives.
5. Click Browse.
6. Select a location to save the report:
7. Review the generated report.
   - Contents of the Excel or HTML Project Report
   - Contents of the Maps Report

**Contents of the Excel or HTML Project Report**

The HTML version of the project report is a set of Web pages saved in a group of subfolders, and the Excel version is a single Excel file with a number of sheets. The HTML and Excel versions contain the same information.

An HTML project report for a TD project is shown below.
The Excel version of the same project report is shown below.
In both formats, the following information is generated:

<table>
<thead>
<tr>
<th>HTML Page or Excel Sheet</th>
<th>Contents for TDD Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Summary</td>
<td>Link to each page or sheet in the report</td>
</tr>
</tbody>
</table>
### HTML Page or Excel Sheet

<table>
<thead>
<tr>
<th>Data Set Type</th>
<th>Contents for TDD Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source data set</td>
<td>•</td>
</tr>
<tr>
<td>Target data set</td>
<td>• Data set classification</td>
</tr>
<tr>
<td>Staging data source</td>
<td>• Logical tables in the data set</td>
</tr>
<tr>
<td></td>
<td>• Physical tables in the data set</td>
</tr>
<tr>
<td></td>
<td>• Data objects in the data set</td>
</tr>
<tr>
<td></td>
<td>• JDBC connection names and types</td>
</tr>
<tr>
<td></td>
<td>• Tables in each connection</td>
</tr>
<tr>
<td>Source tables</td>
<td>•</td>
</tr>
<tr>
<td>Target tables</td>
<td>• Table classification</td>
</tr>
<tr>
<td>Staging data source tables (one entry per table)</td>
<td>• Data set name</td>
</tr>
<tr>
<td></td>
<td>• JDBC connection name</td>
</tr>
<tr>
<td></td>
<td>• Number of rows</td>
</tr>
<tr>
<td></td>
<td>• Column names</td>
</tr>
<tr>
<td></td>
<td>• Column Analysis data</td>
</tr>
<tr>
<td></td>
<td>• Parent tables to this table</td>
</tr>
<tr>
<td></td>
<td>• Joins with physical table</td>
</tr>
<tr>
<td></td>
<td>• Mapping with physical table</td>
</tr>
<tr>
<td>Source data objects</td>
<td>• Root table</td>
</tr>
<tr>
<td>Target data objects</td>
<td>• logical table names and aliases used in the data object</td>
</tr>
<tr>
<td>(one entry per data object)</td>
<td>• logical table relationships within the data object</td>
</tr>
<tr>
<td>Maps (one entry per map)</td>
<td>• Logical source and target tables in the map</td>
</tr>
<tr>
<td></td>
<td>• primary and alternative joins with statistics</td>
</tr>
<tr>
<td></td>
<td>• binding conditions with statistics</td>
</tr>
<tr>
<td></td>
<td>• transformations with statistics</td>
</tr>
<tr>
<td></td>
<td>• alternative transformations (optional)</td>
</tr>
<tr>
<td></td>
<td>• transformations sorted by type and by confidence level</td>
</tr>
<tr>
<td></td>
<td>• join, binding, and transformation that each source column is used in</td>
</tr>
<tr>
<td>Data Rules (one entry per data rule)</td>
<td>• Data rule type</td>
</tr>
<tr>
<td></td>
<td>• constraining type</td>
</tr>
<tr>
<td></td>
<td>• whether data rule is shared</td>
</tr>
<tr>
<td></td>
<td>• names and data types of source, target, data rule, and description columns</td>
</tr>
</tbody>
</table>

---

**Contents of the Maps Report**

The Maps report formats information about all maps in the project in a read-only grid. All maps are shown, even ones that are incomplete or cannot be exported.

You can filter the rows and rearrange or hide columns.

After reviewing the grid, [Export the Maps Report as a .csv File](#).
See the Grid Panes description for common actions you can perform in this grid.

The columns in the report are described below. Columns marked with * contain map-level information and are only populated for the first row in the map.

- **Source Column**: the source column used in the transformation, in the format `<datasourcename>.<schemaname>.<tablename>.<columnname>`
- **Source Business Term**: the source column Classification (Group Names)
- **Target Column**: the target column used in the transformation, in the format `<datasourcename>.<schemaname>.<tablename>.<columnname>`
- **Target Business Term**: the target column Classification (Group Names)
- **Transformation Rule**: the source-to-target transformation expression. This is blank when the transformation is a copy-over of the source column. Null and constant transformations are listed but their corresponding Source Columns are empty.
- **Transformation Function**: empty column for IBM InfoSphere DataStage use
- **Annotation**: any Notes entered for the transformation
- **Join**: empty column for IBM InfoSphere DataStage use
- **Filter**: the primary Where clause, if any, for the map
- **Specification Description**: the map name and the primary join expression, if any
- **Status**: the Approved status of the transformation. If the transformation is unapproved, this is the status of the Transformation step.
- **Last Update Description**: any Notes entered for the project in the Home tab. If no project notes exist, this is the project name.
- **Last Update**: the Last Modified date for the project in the Home tab
- **ID**: the target Column Number
- **Editor**: the currently logged-in Discovery Studio user.

### Export the Maps Report as a .csv File:

This procedure describes how to export the Maps report as a .csv file.
About this task

Any map in the report will not be included in the .csv export if it contains:

- any logical tables that use joins, transformations, renamed columns, or other complex expressions.
- self-joins (table aliases).

To export this report as an IBM InfoSphere FastTrack-compatible .csv file:

Procedure

1. Select the columns and rows for export:
   - To export only the visible rows and columns (except maps that cannot be exported, as described above), select **Export All Maps**.
   - To export only a subset of the visible rows, such as a single map (except maps that cannot be exported, as described above), select the desired rows by clicking and dragging or using CTRL-click, then select **Export FastTrack Maps**.
   - To export all rows and columns in the report (except maps that cannot be exported, as described above), even if some rows or columns have been hidden, select **Export FastTrack Maps**.

2. Specify an export directory.

3. Click OK.

   One .csv file is created per map, using the map name as the filename. The full contents of the rows are exported exactly as displayed onscreen.

   If a .csv file of the same name already exists in the destination directory, it will be overwritten unless the file is in use by another application.

Table Report

The Table report contains information about how the source and target tables map to one another, based on the transformations Discovery has discovered so far. These statistics improve as you refine the maps and transformations, and can be thought of as progress reports providing a snapshot of mapping completion.

Note: The Table Report includes mapping statistics and results based on transformations only. Statistics and results from Where clauses and Join conditions are not shown here.

This report can be generated as a database table or as a read-only report. The read-only report cannot be saved.

Generate the Table Report

Generating a report from a large project can take 30 minutes or longer, and is memory-intensive. To provide the largest amount of free memory for this task, restart Discovery Studio and then immediately generate the report before performing any other actions.

Procedure

1. Open a project.

2. From any tab, select **Tools>Reports>Table**. The following dialog appears.
3. Select either DBTable or Grid.
4. Select the filepath.
   • For Grid, click OK.
   • For DB Table, enter the name of a valid schema, then click Save. The table is created with the project name.

**Contents of the Table Report**
Discovery Studio generates a full report in the selected format. For DB Table, the new table is saved in the staging data source. These reports contain the same information and have the same layout. The Grid format is shown below.

**About this task**

The Source tables are listed on the left, and the Target tables are in the middle. Map information is on the right. Tables in the same row map to each other, and the statistics indicate the degree of mapping so far. If a table does not map at all yet, it is listed in a row by itself with a NOT MAPPED status.

See the Grid Panes description for common actions you can perform in this grid.

The columns in the report are described below.

- **Source Dataset**: name of the source data set
- **Source Datasource**: name of the source data source
- **Source Table**: name of the table or text file in the source data source
- **Source Transformation Stats**: how many rows in this table have been mapped to a target row, presented as both an absolute number and as a percentage of the total number of columns in the table
• **Source Status:**
  - **Not Mapped** - none of the columns are referred from any transformation expressions generated by the corresponding map
  - **Partially Mapped** - some but not all columns are used in some transformation expressions generated by the corresponding map
  - **Mapped** - every column is used in a transformation expression generated by the corresponding map

• **Target Dataset:** name of the target data set
• **Target Datasource:** name of the target data source
• **Target Table:** name of the table or text file in the target data source
• **Target Transformation Stats:** how many rows in this table have been mapped to a source row, presented as both an absolute number and as a percentage of the total number of columns in the table

• **Target Status:**
  - **Not Mapped** - none of the columns have non-null transformation expressions in a target map
  - **Partially Mapped** - some but not all columns have non-null transformation expressions in this target map
  - **Mapped** - every column has a non-null transformation expression in at least one target map

• **Map Approved Status:** false (you have not checked the Approval Checkboxes for this map) or true (approval checkbox is checked)

• **Map Name:** name of the map that contains this particular source-target mapping

The **Transformation Stats** columns display the same data each time a particular table is listed. This can result in some redundant values in the grid, because a given table is listed more than once if it belongs to more than one map.

**Column Report**

The Column Report displays information on the mapping between the individual columns of each source table and each target table. Each source column is listed, whether or not it maps to a target column yet. The only target columns listed are those that have mappings to a source column.

Each row represents a column-to-column correlation. (Source columns without target or map data do not yet map to a target column.) This report displays, for each correlation, the row and value hit rates, the map through which this column relationship was discovered, and the SQL expression representing the relationship between this source-target column pair.

These statistics improve as you refine the maps and transformations.

This report can be generated as a database table or as a read-only grid. The read-only grid cannot be saved.

**Generate the Column Report**

Generating a report from a large project can take 30 minutes or longer, and is memory-intensive. To provide the largest amount of free memory for this task, restart Discovery Studio and then immediately generate the report before performing any other actions.
Procedure
1. Open a project.
2. From any tab, select Tools>Reports>Column. The following dialog appears.

3. Select the appropriate format, DBTable or Grid.
   - For Grid, click OK.
   - For DB Table, enter any valid schema name and click OK.

Contents of the Column Report

Discovery Studio generates a full report in the selected format (database table or view-only grid). These reports contain the same information and have the same layout. The Grid format is shown below.

The Source tables are listed on the left, and the Target tables are in the middle. Map information is on the right. Tables in the same row map to each other, and the statistics indicate the degree of mapping so far.

See the Grid Panes description for common actions you can perform in this grid.

The columns in the report are described below.
- Source Dataset: name of the source data set
• **Source Datasource**: name of the source data source
• **Source Table**: name of the table or text file in the source data source
• **Source Column**: name of the source column
• **Target Dataset**: name of the target data set
• **Target Datasource**: name of the target data source
• **Target Table**: the table or text file in the target data source
• **Target Column**: a target column that maps to a source column
• **Row Hit Rate**
• **Transform Name**: name of the map that contains this particular source-target correlation
• **Transform Expression**: SQL expression that transforms the source data into the target data

**Generate BC Report**

**About this task**

This option becomes active when any map is selected. It generates an Excel report listing all rows in the source and target tables in a map, indicating which rows are bound and unbound. The report is saved as a comma-delimited (.csv) file, with the map name used as the default filename.

---

**Create Validation Jobs**

Exception Manager is a separate IBM product that runs in a web browser. On an ongoing basis, Exception Manager monitors data and validates the binding conditions and transformations that were discovered in a TD project.

**About this task**

In order to use Exception Manager, you need to define a validation job for the project in Discovery Studio. This job identifies which expressions in the project are to be validated, how some exceptions will be handled, and how the validation tables will be staged. Each job is assigned a unique job ID, and also contains the unique ID of the Discovery project.

You also need to create a separate .jar file (executable) that contains the unique job ID and the network location of the Exception Manager Server associated with the Discovery installation. This allows the .jar file to be placed anywhere in the network and still trigger the validation job.

**Note**: The only Exception Manager-related actions performed in Discovery are creating the job and the executable, and manually running the job to test it. The executable is not scheduled or monitored from Discovery, and the results cannot be viewed or accessed in Discovery.

In normal operation, you create validation jobs and executables for any number of Discovery projects. A third-party script is scheduled to execute each .jar file on a regular basis, which triggers the corresponding validation job to run. Operators review the results of each validation run in Exception Manager and assign corrective actions as needed.

This chapter describes how to create and update validation jobs and executable .jar files. For more information, see the *IBM InfoSphere Exception Manager User Guide*.  

---

---
About the Validation Job and Executable

You create the validation job in Discovery after finishing a TD project. The validation job is saved in the Discovery repository.

About this task

You can create the executable at the same time as the job, or you can create it later. The executable is a separate file that can be moved to any host in the network that has access to the Discovery Server and Discovery Engine Service.

A third-party script usually triggers the executable on a schedule. Each time the executable is run, the selected expressions in the specific project are validated and results are available in Exception Manager. This allows Exception Manager users to quickly identify data errors and notify appropriate personnel to correct the problems.

Prerequisites for Creating a Job

The Validation Jobs menu option is only available in Transformation Discovery Projects. Also, the Discovery installation must include Exception Manager. If you have a Discovery-only license, you cannot create validation jobs.

The project must contain at least one valid binding condition, transformation or PF Key. In addition, you should select a Primary Key Column for any tables in the project that will use the Incremental With Caching staging mode.

Notifying Exception Manager Users About Changes

It is very important to notify Exception Manager users when you modify a project that is being monitored. Changes to the project can result in incorrect exception results.

Changes to the Job Definition Only

When you modify an existing job definition in the Tools>Validation Jobs menu, you do not need to create a new executable. Notify the Exception Manager steward.

About this task

The steward needs to use the Exception Manager Clean Up Utility to delete the current job exceptions, and then rerun the job.

Changes to the job definition include validating more or fewer maps, selecting different expressions for validation, and so on.

Changes to Expressions Being Monitored

About this task

If an expression is monitored by Exception Manager, changing the expression in Discovery Studio can cause incorrect exception results in Exception Manager. This includes selecting a new primary transformation or PF Key, modifying an existing primary transformation, changing a join condition, and so on.

You do not need to change the job definition or executable, but you should notify the steward, who may or may not need to delete the exceptions.
• If the change is minor, such as changing a transformation expression from `TableA.ColA=TableB.ColA` to `TableA.ColA=TableB.ColA+1`, the steward does not need to take action but should watch for unusual exceptions. However, Exception Manager should be able to reconcile the results without a problem.

• If the change is major, such as changing `TableA.ColA=TableB.ColA` to `TableA.ColA=TableC.ColA`, the steward needs to use the Exception Manager Clean Up Utility to delete the current job exceptions, and then rerun the job.

Changes to the Metadata
When you change the metadata in a project that is monitored by Exception Manager, you need to create a new validation job and a new executable. Notify the Exception Manager stewards and viewers monitoring the job, along with the admin and operator.

About this task
Changes to the metadata include re-importing tables with more or fewer columns, changing the source or target tables in a map, modifying data types, and so on.

Exception Manager users need to perform the following actions:
• A steward needs to delete the old project using the Exception Manager Clean Up Utility.
• The admin needs to add the new project to the Available Projects list for all appropriate stewards and viewers.
• The stewards and viewers need to add the new project to their Watched Project lists.
• The operator needs to monitor the new project.

About Deleting Monitored Projects
To delete a project that is monitored by Exception Manager:

About this task
Procedure
1. Delete the project in Exception Manager using the Exception Manager Clean Up Utility.
2. Delete the project in the Discovery Studio Home tab.

Results
If you do not follow this order, the Exception Manager administrator will not be able to delete the project from Exception Manager.

Create a Validation Job
About this task
To create a validation job and .jar executable file for use with Exception Manager:

Procedure
1. Open a completed TD project containing at least one binding condition, transformation, or PF Key.
2. Select the Tools>Validation Jobs menu option.
Note: If Exception Manager was not installed with your instance of Discovery, this menu option is not visible.

The **Job Configuration** dialog appears.

3. Click the + icon.

The **Properties** dialog appears. The **General** tab is selected by default.
4. In the **General** tab of the **Properties** dialog, enter a name for the new job. Optionally, enter a **Description**.

**Results**

Continue with the **Set Validation Options** procedure.
Set Validation Options
Procedure
1. In the Rules to Validate section, expand the Maps entry to view the maps in the project.

2. Expand the first map. Note that there are two subentries: the map’s binding condition, and a collapsed list of transformations.

3. Set the validation options for the first map’s binding condition:
   • To have Exception Manager validate the binding condition, check the box in front of the binding condition expression. Unchecked binding conditions will not be validated.
   • Change the binding condition’s Exception Limit and Ignore Exceptions options, if needed.
     Exception Limit: The maximum number of exceptions in this particular expression that will be saved in the Exception Manager repository. If more exceptions than this are discovered for this expression, the following occurs:
     – The additional exceptions are discarded.
     – Old exceptions (discovered in the previous validation) are discarded.
     – The exceptions discovered in the new run are displayed as new and not compared to the old exceptions to see if any were repeated.
     Ignore Exceptions: When this box is checked, Exception Manager will still find exceptions in this object (binding condition, transformation, or PF Key), but all exceptions will be marked as Ignored. The object will not be listed in the Dashboard as containing exceptions. However, you can still view the object’s exceptions in the Job Execution Summary.

4. Expand the transformations for the first map.
5. Set the validation options for the first map’s transformations:

   - To have Exception Manager validate the transformation during each run, check the box in front of the transformation. Unchecked transformations will not be validated. You can check the box in the Transformations (Column Name : Expression) row to select or deselect all transformations at once, and then check or uncheck individual transformations.

   - Change each transformation’s Exception Limit value, if needed.

     To set a single Exception Limit value that applies to all transformations in the map, click in the Exception Limit field in the Transformations (Column Name : Expression) row, enter the new value, and then click the arrow that appears in the field. The value is applied to all transformations in the map.

   - Change each transformation’s Ignore Exceptions option, if needed.

6. Set validation options for the binding conditions and transformations for the remaining maps in the project (Step 2 through Step 5).

Set Validation Options for Tables and PF Keys:

Procedure

1. When you have set validation options for all binding conditions and transformations in the project, expand the Tables entry. There is one Tables entry per project.
2. Expand the first table in the list to display all of its PF Keys. In the following example, the first table has one PF Key.

3. Set the validation options for the first table’s PF Keys:
   - To have Exception Manager validate the PF Key, check the box in front of the PF Key. Unchecked PF Keys will not be validated. You can check the box in front of the table to select or deselect all of its PF Keys at once, and then check or uncheck individual PF Keys.
   - Change each PF Key’s Exception Limit and Ignore Exceptions options, if needed.

4. Expand the remaining tables in the list and set validation options for each one’s PF Keys.

5. If the project uses data rules, it is recommended that you check the box to reload each data rule in the project. This refreshes the data rules in case they have been changed in Discovery Studio.

6. For each data rule that is selected for reloading, determine how the updates should be processed.
   - When **Incremental Update** is checked, new values are inserted into the data rule table.
• When it is not checked, all existing data in the data rule table is deleted and replaced with the new data.

7. If all selected tables will use the default (Full) staging options and no scripts are needed, you are ready to [Generate a jar Executable]. Otherwise, use one or both of the following procedures to configure the job as needed:
   • [Change the Staging Options]
   • [Create Scripts]

**Change the Staging Options**
You can change the staging options in an open job definition.

**Procedure**

1. Display the [Staging Options] tab.

2. Expand the [Source Tables] list. You may need to scroll to see all of the source tables in the list.

   **Note:** At least one binding condition, transformation, or PF Key must be selected for validation in the [General Tab] in order for these lists to be populated.

3. Select the first table in the list.
4. Choose the appropriate staging mode and sample set. This determines how Exception Manager will update and store the table's validation results in the Exception Manager staging data source.

**Note:** If the Incremental With Caching option is not available, the table does not have a Primary Key Column selected. If it is important to use incremental caching in this table, save and close the Job Configuration dialog, then return to Discovery Studio and select a Primary Key Column in the Data Sets tab.

- **Full Table:** The selected sample set is validated in each validation run. Rows that generate exceptions are stored in the Exception Rows Cache in the Exception Manager staging data source.
  
  If the only Sample Set listed is “Full Set”, the table does not have any additional sample sets defined. (This dialog lists all sample sets for the table, not just the primary.) If it is important to use a different sample set when validating this table, save and close the Job Configuration dialog, then return to Discovery Studio and create a new sample set in the Data Sets tab.
**Incremental With Caching**: Before each run, Exception Manager will use the selected sample sets to insert, update, or delete rows from the previous run's data in the staging data source and Exception Rows Cache. After validation, the data from the new run replaces the data from the previous run in the staging data source. In addition, rows that generate exceptions are stored in the Exception Rows Cache.

If the only Sample Set listed is “Full Set”, the table does not have any additional sample sets defined. (This dialog lists all sample sets for the table, not just the primary.) If it is important to use a different sample set when validating this table, save and close the Job Configuration dialog, then return to Discovery Studio and create a new sample set in the Data Sets tab.

You only need to select a sample set if you want the particular action performed. Selected sample sets are used as follows:

- **Changed Data Sample Set**: Before executing the run, Exception Manager uses this sample set to insert and update (upsert) the rows in the existing staging table or Exception Rows Cache.
- **Inserted Data Sample Set**: Before executing the run, Exception Manager inserts the rows in this sample set into the existing staging table or Exception Rows Cache.
- **Updated Data Sample Set**: Before executing the run, Exception Manager updates the rows in the existing staging table or Exception Rows Cache with the rows in this sample set.
Deleted Data Sample Set: Before executing the run, Exception Manager deletes any rows in the existing staging table or Exception Rows Cache that match the rows in this sample set.

5. Select the remaining tables in the list and configure their staging modes and sample sets.

6. When you are finished configuring the staging modes and sample sets, Create Scripts if needed, and then Generate a .jar Executable.

Create Scripts
You can create pre-execution and post-execution scripts for an open job definition.

Procedure
1. Click the Advanced tab.
2. Define one or both scripts using JavaScript. The scripts can use any variables you defined. You can also write the script elsewhere and paste it into the appropriate field.
If variables are defined for this project, click **Show Variables** to display them. Existing variables can be used in the script. (See [Define Project Variables](#)).

3. **Generate a .jar Executable**

**About this task**

You must create a .jar executable in order for Exception Manager to run a job. It is not necessary to create the executable when you first define the job; you can save the job definition and create the executable at a later time.

**Note:** You can execute the job from Discovery Studio without creating a .jar file (for example, to test the job). Exception Manager is required in order to view the results. See [Execute a Validation Job Without a .jar File](#).

**Procedure**

1. In the **Executable Path** field of the **General** tab, enter a fully qualified filepath and filename for the executable. Or, click the **...** button to navigate to a directory, then enter the filename.

   The following restrictions apply to the filepath and filename:
   - The executable filename must end in `.jar`.
   - The filepath must contain the fully qualified hostname of the destination host.
   - You can overwrite an existing .jar file. A popup window will ask you to confirm the overwrite.
   - You can save the executable in any location in the network. It can be moved to a different network location later.
2. Click Generate to create the .jar file.

Discovery Studio saves the job definition, then creates the executable .jar file. Exception Manager will use the .jar file and the job definition to validate the project using the options configured here.

Discovery automatically includes the fully qualified hostname of the Exception Manager Server in the executable. This means you can move the .jar file to any directory or host in the network that has access to Exception Manager, and Exception Manager will still run the job.

3. Click OK to save the job and close the Job Properties dialog. The validation job is saved in the Exception Manager staging data source.

**Update a Validation Job**

**About this task**

Before updating a validation job, see Notifying Exception Manager Users About Changes.

To update a validation job:

**Procedure**

1. In Discovery Studio, open the project that contains the job.
2. Select the Tools>Validation Jobs menu option.

   **Note:** If Exception Manager was not installed with your instance of Discovery, this menu option is not available.

   3. Double-click the job name in the Name list of the Job Configuration dialog.
      
      The Properties dialog appears. The General tab is selected by default.

   4. Continue with one or more of the following procedures, as needed:
      
      - Set Validation Options
      - Change the Staging Options
      - Create Scripts

   5. When you are finished making changes, Generate a .jar Executable

**Execute a Validation Job Using a Script**

You can write a batch file to facilitate logging and error handling for automatically-executed validation jobs.
As an example, the following code fragment executes a specific validation job, redirects messages and errors to new files, and performs a predetermined error handling action if the job fails.

```java
java -jar %1.jar 1>%1.out.log 2>%1.err.log else errorlevel -1 <error handling>
```

- `%1.jar` is the validation .jar file executable
- `%1.out.log` is a new text file containing output messages
- `%1.err.log` is a new text file containing error messages
- `<error handling>` is the action to take if an error occurs during job submission.

For example, assume the code fragment above is saved in a script named `runvaljob.bat`, and the error handling is `echo ERROR`. The script fragment looks like this:

```bash
java -jar %1.jar 1>%1.out.log 2>%1.err.log else errorlevel -1 echo ERROR
```

To submit a validation job named `HQ_Project.jar`, you would execute the following command:

```
runvaljob HQ_Project
```

The output is saved in `HQ_Project.out.log`, and errors are saved in `HQ_Project.err.log`. If the job submission fails, the error is displayed.

### Execute a Validation Job Without a .jar File

**About this task**

Validation jobs are usually run by triggering the .jar executable outside of Discovery Studio using a script or some other method. However, you can also execute a job in Discovery Studio without creating a .jar file first. This can be useful for immediately testing the job. You still need Exception Manager in order to view the results.

**Note:** The *IBM InfoSphere Exception Manager User Guide* contains full instructions for executing jobs and interpreting results.

To execute a job from within Discovery Studio:

**Procedure**

1. In Discovery Studio, open the project that contains the job.
2. Select the Tools>Validation Jobs menu option.

   **Note:** If Exception Manager was not installed with your instance of Discovery, this menu option is not available.

   The Job Configuration dialog appears.
3. Double-click the job name in the Name list of the Job Configuration dialog.

   The Properties dialog appears. The General tab is selected by default.
4. At the bottom of the window, click **Run Now**.

**Results**

The job is immediately submitted.

*Note:* A .jar file executable is not created for the job when it is run this way. To run the job outside of Discovery Studio, create an executable.
Chapter 6. Reference

Put your short description here; used for first paragraph and abstract.

reference container for UG reorg

Reference of Grid Column Names

Most columns that appear in the Discovery Studio grids are described below, in alphabetical order. Columns unique to a particular screen are described in that screen’s section.

#

For TD Projects: number of this column in the original database table or text file, from left to right, starting with 1.

For SDD Projects: number of this column in the target table from left to right, starting with 1, determined by the order in which the column was added to the target table. You can drag target columns to renumber their order.

Activity

Number of tasks in the project that are currently queued or running.

Alternative Count

Displays all PF Keys between these two tables. Expand the entry to view the primary PF Key and any alternatives.

Approved

User-defined status that indicates an item (row, data set, expression, etc.) has been reviewed or verified and is considered acceptable. This field is a convenient way to keep track of progress. On some screens, the field is read-only.

In some screens, an Approved item is not rediscovered or recalculated when the task is re-run (this is noted on each screen's Re-Run Steps description).

Approved Columns

Number of target table columns that have been marked as Approved in the Target Table Schema Grid, shown as a percentage of the total number of columns in the target table.

Average Group Size

Average number of matching rows in each group, for this matching condition.

Blank Count
(available from Column Chooser) Number of rows in the column that are blank (empty).

Bound Rows Passing Filter

The subset of rows to which the binding condition applies, given the specified filter.

Cardinality

The number of unique values in a column with the current primary sample set. The most common value in a column is called the **Mode**.

**Note**: Null and empty values are not included when calculating cardinality.

Cardinality is calculated on an individual column basis and is not the result of comparison to another column. This value is never greater than the total number of rows in the column, and is used in calculating several other statistics.

For example, assume the following columns:

<table>
<thead>
<tr>
<th>C1</th>
<th>C2</th>
<th>C3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>E</td>
<td>A</td>
<td>null</td>
</tr>
<tr>
<td>A</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>G</td>
<td></td>
</tr>
</tbody>
</table>

C1 has a cardinality of 5, because the column contains five unique values.

C2 has a cardinality of 2.

C3 has a cardinality of 6 (the null value is excluded).

Cardinality Ratio

For the bound rows, the number of unique values in the source column over the number of unique values in the target column, including null values. This value is also displayed as a percentage.

CDE

An optional designation that allows you to flag one or more specific columns as **Critical Data Elements (CDEs)**. Some subsequent screens display the CDE status, allowing you to quickly sort or filter the grids to identify those columns in later processing. The designation has no effect on discovery or processing.

Columns are marked as CDEs in The Column Analysis Window and in the Column Overlaps window.
The CDE designation can be used in conjunction with the **Classification (Group Names)** field.

**Classification**

The classification of the column or relationship.

In the Column Analysis screen, you can enter a string in each row to help identify or categorize the columns in each logical table. Several subsequent screens display the **Classification** column and allow you to sort or filter the grid based on classification. Groups can be used in conjunction with CDEs.

In the Column Analysis screen, to assign the same name to another row in the grid, click in the **Classification** column for that item. A dropdown menu allows you to select from all classification names used in the grid.

After entering or selecting a classification name, press Enter or tab out of the cell to commit the change.

**Classification Approved**

Indicates whether the Classification has been approved.

**Classification Matches**

All columns in the project with classifications that exactly match the target column.

**Name Matches**

All columns in the project with column names that exactly match the target column.

**Column**

The name of the column in the original logical table.

**Column (Mapped Table)**

**Column (Source Table)**

Ratio of (target table columns with conflicts) to (total number of target table columns).

**Column Expression**

Current SQL expression for the column. The expression can be a constant or any valid SQL expression. Column expressions have the following restrictions:
- Must include the table name (as in `W_DEPTS.CITY`)
- Can only contain columns from the physical tables in the **From** statement.
- Cannot contain other columns or logical columns.

**Column Name**

Name of the column in the original table or text file.
Column Number

Number of this column in the original database table or text file, from left to right, starting with 1.

Columns with Conflict

Number of columns in each group that have conflicts.

Completeness Rate

Ratio of (all values from transformation from Data Set) divided by (number of non-null values).

Conflict Detection Rule

A SQL condition, automatically created by Discovery when a target column is created, to determine how consistent the column values are within a group.

Conflict Resolution Rule

A SQL condition, automatically created by Discovery when a target column is created, that is used to compare two records from the same group during the Merge task. This determines, for each target column, which value is selected as the preferred value during the Merge process.

Confidence Index

Average percentage of the column-level Conflicts Rate, also expressed as (sum of all column level Conflicts Rate(s) divided by (number of columns). This value is one of the key characteristics of matching condition quality.

Conflict Count

Total number of conflicts for all columns in the group.

Conflict Count for <column>

The number of conflicts per group. The grid contains one entry for each target column.

Conflicts Rate

Ratio of (number of groups with conflicts for this column) to (total number of groups).

Correctly Transformed Rows (Hits) Passing Filter

The subset of correctly transformed bound rows to which the given filter applies.

Data Rule

The data rule, if any, associated with the target match.

Data Set
Name of the column’s data set.

Data Set (Mapped Table)

The data set that the mapped column belongs to.

Data Set (Source Table)

The source table’s data set.

Data Sets (Unified Schema)

Each data set in the project is shown, whether or not a column from that data set is currently involved in the target table.

Data Sets (Home tab)

The number of data sets in the SDD project.

Data Type

The target column’s data type, either read from the source or discovered by Discovery. Click in the desired cell to select a different datatype from the dropdown list.

TD projects only: If you change the datatype to NumberString or DateTimeString, you also need to enter a definition in the Formats column.

DateTimeString

The column’s data type. Can also contain its precision, scale, and length.

De-duplication Rate

For the selected matching condition, the number of rows that matched (were duplicates) given the current matching condition, divided by the number of original rows. A de-duplication rate of 0% means that none of the rows were duplicates given this matching condition, so no rows were removed. A rate of 100% means that all rows matched given this matching condition, so all but one row was removed. The number of removed rows is equal to (number of aggregated rows) - (Group Count).

Details

Displays all rows to which the selected record belongs to. If Merge processing has been performed, this displays the merged master record that Discovery created for that row, using the default Conflict Resolution Rules.

Exception Limit

The maximum number of exceptions in this particular expression that will be saved in the Exception Manager repository. The default is 1000 exceptions, and there is no maximum limit. If more exceptions than this are discovered for this expression, the following occurs:

• The additional exceptions are discarded.
Old exceptions (discovered in the previous validation) are discarded.
The exceptions discovered in the new run are displayed as new and not compared to the old exceptions to see if any were repeated.

**Expression**

The selected expression from the grid. This expression is applied to the data shown in the window.

**Expression (Show All PF Keys)**

The PF Key expression.

**Filter**

The SQL expression of the where clause (filter). All known filters from previous transformation filter discovery or binding filter discovery.

**Format**

Format types are listed only for NumberString and DateTimeString columns. For all other data types, this column reads N/A (not applicable).

For NumberString or DateTimeString data type formats, this column contains examples of data formats found in the original table or text file column. Discovery will only use the formats listed here for the data in each column. Formats are applied to the data in the order shown.

You can add more formats for each column. Click on a format type to Change the NumberString Format or Change the Date, DateTime, or DateTimeString Format.

**Group Consistency Rate**

The ratio of good groups (without conflicts) to all groups (with and without conflicts). A group has a conflict if at least one values pair from different rows of the group does not satisfy the Conflict Detection Rule. This value is a helpful indicator of the amount of cleanup that may be required in the data.

**Group Count**

The total number of groups generated by this matching condition. Each group contains at least one target table row.

**Group Number**

A unique number assigned to each group for tracking purposes. Group numbers are not always sequential.

**Group Size**

The number of records in the group.

**Hits (Reverse Pivots Tab)**

The total number of rows in the Target Column (Reverse Pivot) that are accurately derived from the Source Column(s), when the current Transformation is applied.
Hits

The percentage of rows in the target column that are accurately derived from the Source Column(s), when the current Transformation is applied.

Ignore Exceptions

When this box is checked, Exception Manager will still find exceptions in this object (binding condition, transformation, or PF Key), but all exceptions will be marked as Ignored. The object will not be listed in the Dashboard as containing exceptions. However, you can still view the object's exceptions in the Job Execution Summary.

Incorrectly Transformed Rows (Hits) Passing Filter

The subset of incorrectly transformed bound rows to which the given filter applies.

Last Modified

Date and time that any object in this project was last saved.

Length

Character length of the column.

Limitations on the length, precision, and scale for Varchar data types are shown below.

Table 23. Varchar data types: length, precision, and scale

<table>
<thead>
<tr>
<th>Database type</th>
<th>Varchar Length (Min to Max)</th>
<th>Varchar Precision (Min to Max)</th>
<th>Varchar Scale (Min to Max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>1 to 4000</td>
<td>1 to 38</td>
<td>0 to 38</td>
</tr>
<tr>
<td>DB2</td>
<td>1 to 4000</td>
<td>1 to 31</td>
<td>0 to 31</td>
</tr>
</tbody>
</table>

- For database tables, the column length is defined in the metadata.
- For text files, the Numberstring max length is 38 for an Oracle staging data source or 31 for a DB2 staging data source, regardless of the actual length of data in the column in the original table.

In some grids, you can left-click in the cell and type a new value, or click the up and down arrows to change the maximum length.

For the selected map, the number of unique values in a column. For example, a column containing eight rows, where five of the values are a and the other three are b, has a cardinality of 2.

Map Selectivity

For the selected map, the degree of uniqueness of the values (including nulls) in the column, calculated as Cardinality divided by (row count minus Null Count).

Mapped Count

Number of target table columns that this column has been mapped to.
Mapped Data Sets (Unified Schema)

The percentage of data sets in the project that have at least one non-null column mapped to this target table. A value of 100% means at least one non-null table from each data set is mapped to the target table. The Non-null Columns value indicates how many columns are from each data set.

Mapped Row Count (Unified Schema)

The current row count in the target table.

Mapped To

Name of the target column(s) that this table column is mapped to.

Maps

Number of maps in the project.

Matching Condition

The SQL expression used to compare the rows in the target table.

Max

Largest or greatest value in the column.

Max Group Size

Number of rows in the largest group, given this matching condition.

Max Rows from Source

The most number of rows that a single data set contributes to this group.

Min

Smallest or lowest value in the column.

Min Row Hit Rate with Target

Estimated minimum row hit rate between the source and the target using the suggested transformation, including any columns already mapped

Min Value Hit Rate with Target

Estimated minimum value hit rate between the source and the target using the suggested transformation, including any columns already mapped

Misses (Reverse Pivots Tab)

The total number of rows in the Target Column (Reverse Pivot) that are not accurately derived when the Transformation is applied to the Source Column(s).

Misses (Transformations Tab)
The percentage of rows in the Target Column that are not accurately derived when the Transformation is applied to the Source Column(s).

Mode

Most common value in the column, not including null values. This is calculated only if a particular value appears in more than 5% of the rows.

Mode%

Number of times the mode (most common value) appears in this column, displayed as a percentage of all values in the column.

Name

The name of the column or data set.

Name (Unified Schemas)

The name of the Target Table.

Name Match

When checked, this box indicates that the target column and the source column have an exact name match.

Native Type

Column's native type (specific to database vendor).

Non-null Columns

For each data set, the number of non-null columns that are currently mapped to this target table, shown as a percentage of the total number of columns in the target table.

Non-null Count

The percentage of rows in the column that are not null.

Non-null Selectivity

The degree of uniqueness of all non-null values in the column, calculated as:

Cardinality divided by Non-null Count.

Non Nulls %

Percentage of values in the column that are not null.

Non Null Rows

Number of rows in the selected column that are not null.

Notes
Blank field where you can enter Notes, a description, or comments about the item. Left-click in the cell and enter a description of the column. The logged-in user's name and the current timestamp are automatically added to the note.

Null Count

Number of rows in the column that are null.

Ordinal

Number of this column in the logical table, from left to right, starting with 1.

Origin

The origin of the PF Key. This can be:
- Imported: the PF key was imported as metadata from source data or in a CWM file.
- Discovered: the PF Key was discovered automatically by Discovery Studio during data processing.
- User Defined: the PF Key was manually defined in Discovery Studio.

Overlaps with

Table and column that the given column overlaps with, for that particular suggested transformation.

Precedence

An editable value used in Conflict Detection Rules. Higher Trust Levels produce lower precedence numbers. Lower numbers indicate a higher precedence. See Precedence.

Precision

Overall level of precision of the data, expressed as the total number of places on both sides of the decimal point.

In some grids, you can left-click in the cell and type or use the up and down arrow icon to change the overall level of precision.

The primary Match and Merge Analysis expression.

Primary Transformation

The primary transformation involving the Target Column (Reverse Pivot).

Project Name

Name of each project of the selected type (SDD or TD) in the current Discovery Server's repository.

Row Hit Rate

Selectivity, Row Hit Rate, and Value Hit Rate are three key statistics used throughout Discovery as indicators of how well two columns relate to each other.
Row Hit Rate is defined as:

number of rows in column 1 that have corresponding values (hits) in column 2, divided by the total number of rows in column 1.

This value is never greater than 100%.

For example, assume the following columns:

<table>
<thead>
<tr>
<th>T1C1</th>
<th>T2C1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
</tr>
</tbody>
</table>

The Row Hit Rate for T1C1 is 60% (3/5), because out of the five rows in T1C1, three of them (A, B and B) have matching values in T2C1. The two instances of value B in T1C1 are counted as separate matches against the value B in T2C1.

The Row Hit Rate for T2C1 is 100% (7/7), because each of its seven rows corresponds to a value in T1C1. Each row in T2C1 with the value A is counted as a separate hit, and the one instance of the value B counts as one hit even though T1C2 has two rows with the value B.

Row Hit Rate - Mapped (Source Mapping Show Details)

a/b, where

- \( a \) = number of times the column name in the Data Set 1 source table appears in the Data Set 2 mapped table
- \( b \) = total number of rows in the column in the Data Set 2 mapped table

Rows

Total number of rows in the table.

Scale

Level of Decimal Precision of the data, expressed as the number of places to the right of the decimal point. In some grids, you can left-click in the cell and type or use the up and down arrow icon to change the level of decimal precision.

Check this box to update the current transformation with the suggested transformation. The change is made when you click OK to close the window.

Selectivity

Selectivity, Row Hit Rate, and Value Hit Rate are three key statistics used throughout Discovery as indicators of how well two columns relate to each other.
Selectivity is the degree of uniqueness of the values (including nulls) in the column, calculated as:

Cardinality divided by (row count minus Null Count)

Selectivity is calculated on each column individually and is not the result of comparison to another column. This value is never greater than 100%.

For example, assume the following columns:

<table>
<thead>
<tr>
<th>Table 25. Example of selectivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>E</td>
</tr>
<tr>
<td>F</td>
</tr>
</tbody>
</table>

The Selectivity for C1 is 100% (5/5), because C1 has five rows and each one contains a unique value.

The Selectivity for C2 is 29% (2/7). In the seven rows of C2, there are only two unique values.

The Selectivity for C3 is 100% (7/7). C3 contains seven rows, with each row containing a unique value.

Selectivity has several variations and related statistics:
- Non-null Selectivity
- Null Count
- Non-null Count
- Source Bound Selectivity
- Target Bound Selectivity
  Selectivity - Mapped (Source Mapping Show Details)

a/b, where
- a = Cardinality of the column in the Data Set 2 mapped table
- b = total number of rows in the Data Set 2 mapped table

Selectivity - Source (Source Mapping Show Details)

a/b, where
- a = Cardinality of the column in the Data Set 1 source table
- b = total number of rows in the Data Set 1 source table

Source

Name of the project's Source data set.

Source Bound Selectivity
Source bound cardinality divided by the row count.

Source Column

The source column associated with the target match or expression.

Source Hits

Number of rows in the source table(s) that will be used in deriving the target table.

Sources Contributed

Number of data sets that are contributing to this group.

Sparse

Indicates whether the column is sparse or not, based on the Mode%. A sparse column contains mostly the same value except for a few exceptions. In some grids, you can check or uncheck the box to manually override the assigned designation.

Sparse columns are treated differently than non-sparse columns during processing because the large percentage of duplicate values makes it more difficult for Discovery to immediately determine whether a column match is a valid relationship.

Table

Name of the column’s logical table.

Table (Mapped Table)

The table that the mapped column belongs to.

Table (Source Table)

Name of the project’s Target data set.

Target Bound Selectivity

Target bound cardinality divided by the row count.

Target Column

Target column for the suggested transformation.

Target Column

Name of the target column involved in the expression.

Target Column (Reverse Pivot)

The name of the pivot column.

Target Column (Target Matches)

The target column associated with the target match.
Target Columns (Unified Schema)

Name of the target column(s) that this table column is mapped to.

Target Hits

Number of rows in the target table that can be derived from the source table(s).

Total Rows Passing Filter

The total number of rows that passed the filter.

Transformation

Transformation, if any, associated with the filter.

Transformation (Unified Schemas)

The SQL expression that associates the selected target column with each data set in the project. Expressions are defined in the Source Mapping Definition Grid. An expression of null means that no table column from that data set is currently mapped to this target column.

Trust Level

A discovered statistic indicating the trustworthiness of the data in a particular data set. The highest value is 1.0, indicating high trust in the data. A higher Confidence Index indicates a lower Precedence.

Type

The type of binding:
- one to one
- one to many
- many to one
- many to many

Type (Correlations View)

The type of fuzzy match that the selected target column has with each listed source column. There are three types of fuzzy matches:
- Overlap: characters that the source and target have in common, up to threshold
- Subset: number of all target characters that are also in the source
- Superset: number of all source characters that are also in the target

Unbound Rows Passing Filter

Unbound rows that passed the filter.

Unique
Whether or not each value in the column is unique. If any value is duplicated, the column is not unique.

Unmatched Row Rate

Ratio of unique rows (rows not matching any other rows in all data sets) to the total number of rows in the aggregated table, given this matching condition.

Value Hit Rate

Selectivity, Row Hit Rate, and Value Hit Rate are three key statistics used throughout Discovery as indicators of how well two columns relate to each other. Value Hit Rate is calculated as:

\[
\text{Value Hit Rate} = \frac{\text{number of unique values in column 1 with corresponding values (hits) in column 2}}{\text{Cardinality of column 1}}
\]

This value is never greater than 100%.

For example, assume the following columns:

<table>
<thead>
<tr>
<th>T1C1</th>
<th>T2C1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>B</td>
</tr>
</tbody>
</table>

The Value Hit Rate for T1C1 is 50% (2/4). There are four unique values in T1C1, and two of those values (A and B) have corresponding values in T2C1. The Value Hit Rate for T2C1 is 100% (2/2), because T2C1 has two unique values (A and B), and both of those have corresponding values in T1C2.

Value Hit Rate - Mapped (Source Mapping Show Details)

\[
a/b, \text{ where} \\
\cdot a = \text{total number of unique column names in the Data Set 2 mapped table that appear in the Data Set 1 source table} \\
\cdot b = \text{Cardinality (the total number of unique values for that column name in the Data Set 2 source table)}
\]

Value Hit Rate - Source (Source Mapping Show Details)

\[
a/b, \text{ where} \\
\cdot a = \text{total number of unique column names in the Data Set 1 source table that appear in the Data Set 2 mapped table} \\
\cdot b = \text{Cardinality (the total number of unique values for that column name in the Data Set 1 source table)}
\]
The sum of the hit rates for all columns, multiplied by the number of bound rows in the tables the map is comparing.

Where Clause

The SQL expression of the where clause.

Reference of Icons and Buttons

This section describes the icons that appear in the step tabs, in grids, and in other places on screen. To discover the name of an icon, hover over the icon with the mouse, and the name appears in a popup.

- Tab Status Icons
- Icons in Grids

Tab Status Icons

The following icons appear in each tab name (except Home) to indicate the status of data processing, discovery, or other work in the tab.

The icons and descriptions are listed below.

Not Run

You have not yet started to work in this step.

Current

The discovery task for this step has been successfully completed.

Approved

You have marked this step as Approved by clicking the Approve Step button. This status is not assigned or verified by Discovery.
Stale

Some data or results in this step may have been affected by activities performed in another step. Reopen this tab and check the data; you may need to [Refresh Statistics] re-run the step or take other action.

Modified

Data in this step has been manually modified after discovery. This status also appears when you click [Un-approve Step], or when you perform any action in a previously approved step.

Completed With Errors

The last processing task for this step was completed with errors.

Failed

The last processing task for this step failed.

Cancelled

The last processing task for this step was cancelled.

**Icons in Grids**

The following icons and buttons appear in most grids. This example shows the Binding Condition grid.
Add

Click this to add a new, blank entry to the grid. See each grid's description for specific instructions.

Advanced Search

Advanced search allows you to search for matching case and matching whole word. You can also search for column names, to quickly find a particular column in a large table. Click and enter the string, select the appropriate options, and click Find. Also see Search.

Alphabetical Sort

Removes category dividers and sorts the entire list in alphabetical order.

Auto Filter Row

Displays the filter bar, allowing you to filter the grid by rows. Only the rows that match the filter criteria are displayed. For more information, see Auto Filter Row.

Back

Displays the next highest view in the Summary Views or Overlaps Views. See Drill Down and Up.
Categorized Sort

Sorts the entire list in alphabetical order within each category.

Column Chooser

The Column Chooser allows you to temporarily hide columns of statistics, or to display additional statistics calculated but not shown by default. For more information, see Column Chooser.

Data Rule

In the Transformation tab, this icon represents a data rule. Click it to display the Data Rules Window.

Delete

Click this to delete the selected row from the grid. A deleted row cannot be undeleted; you must manually re-enter it or re-run discovery if it was a discovered expression. In some grids, you can Shift-click to select consecutive rows to delete at once. Use Ctrl-click to select non-consecutive rows.

Details

Displays all rows in the group that the selected record belongs to. If Merge processing has been performed, the first record displayed is the merged master record that Discovery created for this group, using the current Conflict Resolution Rules.

Error List Display

Display the Error List.

Export Hits, Export Misses
In the Binding Conditions tab, click this icon to **Export Hits or Misses**.

**Export Grid Content**

You can export the contents of a grid to a file in a variety of formats: comma-delimited, tab-delimited, Excel, HTML, or XML. For more information, see **Export Rows**.

**Export Grouped, Export Merged**

Exports the grouped or merged result of Matching or Merging along with group statistics and record identity.

**Filter Matches**

In the Where and Transformation tabs, click this icon to display the **Filter Matches Grid**.

**Focus**

When a row is selected, the **Focus** command filters the rows on the other side to display only the rows that the selected row maps to. In the following screen shot, selecting **Focus** in the parent table affects the rows displayed in the child table.

- To focus, click on a row in one table and then click the **Focus** icon. (You can also just double-click the row.) The rows in the other table are restricted to display only rows that map to the selected row.
• To redisplay all of the rows in the other table, click the **Focus** icon again.

If you try to Focus and nothing happens, the most likely reason is that the source or target columns in the binding condition are not currently displayed in the Data Preview pane. Add the columns into the other pane, then try again.

**Generate Data Rule**

In the Transformation tab, click this icon to **Generate a Data Rule From the Transformations Tab**

**Move Up, Move Down**

When an entry in the Columns or Tables list is selected, click this icon to move the entry up or down in the list. Expressions are applied to the data in the order they appear in the list.

**Notes**

Some of the screens, dialogs, and grids have a **Notes** field where you can enter a description, comment, or other notes about the object. For more information, see **Notes**

**Output Pane Clear**

Delete the contents of the Output Pane.

**Output Pane Display**

Display the **Output Pane**

**Primary Expression Indicator**

**Primary Key**

A key icon by a column name in a diagram indicates the column has some kind of relationship to a column in another table. If more than one key icon is shown, the table has a composite primary key.
Pushpin

The pushpin icon allows you to collapse and expand certain panes. See Autohide Pane.

Redefine Data Rule Table

In the Transformation tab, click this icon for Details: Table Redefinition. For defined data rules, click this icon to define the table.

Refresh Statistics

Statistics are calculated for expressions, formats, and other types of data. When a statistic becomes stale, or obsolete, it is shown in the grid in italic text. Statistics can become stale for several reasons:

- When you modify an expression, the statistics for that expression become stale.
- When you first define (create) an expression, the statistics columns are empty because Discovery has not yet discovered the statistics for the new expression.
- In the Maps sub-tabs, changing an expression can cause related statistics in the same Maps tab or other Maps tabs to become stale.

To refresh or discover statistics for a single expression or format in a grid, click anywhere in the row to select it and then click the Refresh Statistics icon. This discovers or rediscovers the statistics for the expression or format and updates the grid, changing the statistics from italics to normal text.

In some grids, you can Shift-click on consecutive rows to refresh all of the selected rows at once. Use Ctrl-click to select non-consecutive rows, or Ctrl-A to select all rows.

You can refresh all of the statistics in a Maps tab at once by re-running the step and selecting Only Refresh Statistics.
Statistics for invalid expressions or formats cannot be refreshed.

Reimport Data Rule (Import Data Rule)

In the Transformation tab, click this icon to Details: Reimport a data rule. For defined data rules, click this icon to import the data rule.

Search

The Search feature locates a string in the displayed data. After you type a string, press Enter. The results are filtered to display only those with the string.

The arrow displays a dropdown menu with previous Search strings. Also see Advanced Search.

Share Data Rule Table

In the Transformation tab, click this icon to Details: Share Table

Show All Items

This filters the current view to show all columns, to show only columns marked as Critical Data Elements (CDEs) or to show only columns not marked as CDEs.

Show Hits, Show Misses, Show Data
Click this icon to display the hits, misses, or all data in the Data Preview screen.
Other options may be listed, such as Show Duplicates.

Show Hits, Show Misses
Click this icon to display the hits or misses in the Data Preview screen.

Show Overlaps/Show Non-overlaps
Filters the current view to display only the columns that overlap in the selected data set or tables, or to display only the columns that do not overlap.

Target Matches
In the Binding Conditions tab and the Data Rules Window, click this icon to Show All.

Top
Jumps to the top-level view for the selected tab Summary Views or Overlaps Views.

Update Count
Refreshes the total number of rows in the target table, including Null rows.

Word Wrap On/Off
Turn word wrap on or off in the Output Pane

Reference of Task Options
This section describes the task options available in Discovery.
About Options

Options affect specific parameters of Discovery algorithms and queries. The default option set cannot be changed and is available in every project.

Using New Options in Old Projects

When you install a new version of Discovery over an old version, Discovery uses any new options or new default values when you run tasks on the old and new projects.

About this task

However, new options do not automatically appear in the options lists of any old projects. New Option Sets are also not automatically added to an old project.

If you want to use a new option set, or if the default value of a new option is not acceptable for an old project, you need to export the project and re-import it. This refreshes the options list of the old project with all of the current options, default values, and option sets. You can now select new option sets and modify values of new options.

Global Options and Profile Options

The Global options listed below appear in every step. When you make a change to a Profile or Global option, the change is implemented throughout the entire option set.

The Profile options listed below appear in every step except the Maps steps (because profiling is not performed in the Maps steps).
Table 27. Global Options and Profile Options

<table>
<thead>
<tr>
<th>Option Type</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>Column prefix threshold</td>
<td>0.8</td>
<td>Threshold used in discovering column name prefixes to ignore. Every time it imports a table, Discovery examines the first character of each column name, and if the percentage of all columns that share the same first character in common exceeds this threshold, Discovery concludes that it is part of a prefix. It then examines the second character, applies the same logic, and so on until it reaches a character that does not meet this threshold, which it then concludes is not part of a prefix. By this process, it can identify character string candidates for column name prefixes. Discovered prefixes are displayed in the Column Name Prefix field on the Properties tab of the table object.</td>
</tr>
<tr>
<td>Global</td>
<td>Create indexes on all columns of all staging tables</td>
<td>FALSE</td>
<td>Determines whether or not indexes are created on all columns of all tables in the staging data source. This has a minor impact on performance.</td>
</tr>
<tr>
<td>Global</td>
<td>Data rule maximum number of rows</td>
<td>100</td>
<td>Maximum number of rows that can be used in an inline data rule. Data rules with more than this number of rows are saved as a lookup table.</td>
</tr>
<tr>
<td>Global</td>
<td>Database platforms which do not support derived table optimization for logical tables</td>
<td>TERADATA, MSSQL, SYBASE</td>
<td>When the source or target data source uses a database platform listed here, Discovery will always stage the FullSet sample set.</td>
</tr>
<tr>
<td>Option Type</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>Global</td>
<td>Deploy derived table optimization for logical tables</td>
<td>FALSE</td>
<td>True = When the primary sample set of a logical table is not the FullSet sample set, Discovery will attempt to execute queries against the source data source. If that is unsuccessful, Discovery will stage the FullSet sample set and execute the query against that. False = Discovery will stage the FullSet sample set every time it needs to execute a query against a logical table.</td>
</tr>
<tr>
<td>Global</td>
<td>Disable quoting lowercase identifiers in Sybase</td>
<td>TRUE</td>
<td>True = disable quoting the lowercase identifiers in Sybase. False = allow lowercase identifiers to be quoted.</td>
</tr>
<tr>
<td>Global</td>
<td>Enable discovery progress indicator</td>
<td>TRUE</td>
<td>This activates or deactivates a status bar displaying the approximate percent completion of a Maps task in the Activity Viewer. Turning off this option provides a small performance improvement.</td>
</tr>
<tr>
<td>Global</td>
<td>Enable DPI tracing</td>
<td>FALSE</td>
<td>When True, discovery progress indicator trace messages are logged to the Debug Log. When False, these messages are not logged.</td>
</tr>
<tr>
<td>Global</td>
<td>Enable old stats calculation model for Binding Conditions?</td>
<td>FALSE</td>
<td>Change this option only if requested by a support engineer.</td>
</tr>
<tr>
<td>Global</td>
<td>Enable old stats calculation model for filters?</td>
<td>FALSE</td>
<td>Change this option only if requested by a support engineer.</td>
</tr>
<tr>
<td>Global</td>
<td>Enable old stats calculation model?</td>
<td>FALSE</td>
<td>Change this option only if requested by a support engineer.</td>
</tr>
</tbody>
</table>
Table 27. Global Options and Profile Options (continued)

<table>
<thead>
<tr>
<th>Option Type</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>Enable Oracle Join</td>
<td>FALSE</td>
<td>(Applies only to Oracle staging data source and repository)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Uses either standard ANSI joins or Oracle-specific joins for all queries during discovery.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>False: use standard ANSI joins.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>True: use Oracle-specific joins.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Oracle join syntax contains significant restrictions and is not equivalent to ANSI join syntax. If a Discovery task contains failures involving joins across wide tables, change this to True.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>However, when this option is True, SQL errors may occur with a complex SQL expression. Simplify the expression to comply with Oracle restrictions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see <a href="#">Troubleshooting Map Generation</a>.</td>
</tr>
<tr>
<td>Global</td>
<td>Intermediate discovery save interval</td>
<td>15</td>
<td>Defines in minutes how frequently Discovery performs intermediate saves while a task is running.</td>
</tr>
<tr>
<td>Global</td>
<td>Max number of columns for aggregation discovery and cardinality queries</td>
<td>200 (DB2), 500 (Oracle)</td>
<td>Used in queries for aggregation discovery, map cardinality, and target table cardinality. The default value varies depending on the database type used for the staging data source.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Limits the number of columns used simultaneously in certain queries. Querying too many columns simultaneously can result in large SQL statements that are too complex for the staging data source.</td>
</tr>
<tr>
<td>Option Type</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Global</td>
<td>Maximum number of characters to stage in a column</td>
<td>512</td>
<td>The maximum number of characters to stage in a column. Longer values will be truncated during staging. A value of -1 means no maximum (the database limitations are used for the column).</td>
</tr>
<tr>
<td>Global</td>
<td>Maximum number of records to dump when staging fails</td>
<td>-1</td>
<td>When Discovery encounters a bad record during processing, this value determines how many records from the block are dumped to the debug log. Options are: -1: dump all records in the block 0: do not dump any records from the block n (any positive number): dump n records from the block (if this is less than the full block size, the bad record may not be included in the dump)</td>
</tr>
<tr>
<td>Global</td>
<td>Maximum number of select expressions for statistics refresh</td>
<td>100</td>
<td>The maximum number of select expressions that can be grouped together in a single SQL statement for transformation and filter statistics refresh.</td>
</tr>
<tr>
<td>Global</td>
<td>Maximum VM Size for Java Discovery Engine in Megabytes</td>
<td>1000</td>
<td>If a JVM (Java virtual machine) OutOfMemoryError occurs, this limit can be increased to 1200. Do not lower the value below the default. The default value is sufficient for all but the most extreme cases.</td>
</tr>
<tr>
<td>Global</td>
<td>Minimum amount of time between DPI updates</td>
<td>60</td>
<td>This option determines the minimum number of seconds that must pass before the status bar is updated again with the approximate percentage of the task’s completion. For long tasks, the actual updates may be farther apart.</td>
</tr>
<tr>
<td>Option Type</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>Global</td>
<td>Number of Rows</td>
<td>5000</td>
<td>During ODBC bulk reading, this is the number of rows in the record block. For very wide tables, you may need to lower this value to avoid out-of-memory errors.</td>
</tr>
<tr>
<td>Global</td>
<td>Pattern separator characters</td>
<td>^ @#$%&amp;*()+=-=[]{}:&quot;;'&lt;,./'</td>
<td>Characters that are not considered as letters or digits. These characters are shown as is in Pattern Frequency Analysis. Non-printable characters such as tab, line feed, and carriage return (ANSI codes 9, 10, 13) are always considered as token or pattern separators.</td>
</tr>
<tr>
<td>Global</td>
<td>Profiling cache length</td>
<td>20000</td>
<td>Cache length used in profiling.</td>
</tr>
<tr>
<td>Global</td>
<td>Save intermediate discovery results</td>
<td>FALSE</td>
<td>When set to Yes, Discovery incrementally saves the results of running tasks (Column Analysis, etc.) at a configurable interval, until the task is complete. If the task is interrupted for any reason (system failure, crash, etc.) and you re-run the task, it resumes from the point where the failure occurred. When left at the default No, tasks will complete more quickly, but the results will be lost if the task is interrupted before completion.</td>
</tr>
<tr>
<td>Global</td>
<td>Staging Error Threshold</td>
<td>.01</td>
<td>The maximum number of staging errors that can occur before Discovery terminates the staging task, determined by (total number of staged rows) x (value). Enter value as a ratio: .01 defines the maximum as 1% of the rows.</td>
</tr>
<tr>
<td>Option Type</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Global</td>
<td>Token separator characters'</td>
<td>`@#$%^&amp;*()_+={}</td>
<td><code>&lt;&gt;,?</code>/`</td>
</tr>
<tr>
<td>Profile</td>
<td>Profile in parallel</td>
<td>FALSE</td>
<td>Flag that controls whether Discovery will perform Serial or Parallel profiling. When left at the default No, profiling will be performed in serial mode: all tables will be profiled by a single Discovery Engine, one after another. If you change it to Yes, profiling will be parallel: each table will be handled as a separate task, and can be profiled simultaneously by whichever Discovery Engines take them from the work queue.</td>
</tr>
<tr>
<td>Profile</td>
<td>Profile sparse column detect</td>
<td>0.8</td>
<td>Threshold used to identify sparse columns, during profiling; specifies the minimum percentage of identical values in a column, at or above which the column will be considered sparse, and will be disregarded during column match discovery. If a column’s Mode% value is equal to or greater than this threshold, and if the remaining (non-sparse) values are not constant, the column will be considered sparse.</td>
</tr>
</tbody>
</table>
Table 27. Global Options and Profile Options (continued)

<table>
<thead>
<tr>
<th>Option Type</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Profile sparse detect turbo</td>
<td>TRUE</td>
<td>Flag that enables or disables turbo mode for sparse column identification, during profiling. When turbo mode is enabled, Discovery takes certain shortcuts in sparse column detection, to improve processing speed. This will usually detect sparse columns properly, but it does introduce a slightly greater possibility of errors. (You can go to a table’s Profile tab to review each column’s sparseness status, and change it if necessary.) When turbo mode is disabled, Discovery will calculate an accurate number of the most frequently occurring value available for every column. This could be more time-consuming, but is reliable for every column.</td>
</tr>
<tr>
<td>Profile</td>
<td>Automatic Sampling</td>
<td>False</td>
<td>Used with High Cardinality Threshold and Size Reduction Factor. False (default): Use all values in all columns to determine data types and other results. True: Perform automatic sampling on high-cardinality columns to determine data types and other results.</td>
</tr>
<tr>
<td>Profile</td>
<td>High Cardinality Threshold</td>
<td>1000000</td>
<td>Used when Automatic Sampling is True. Estimated cardinality that a column must have in order to be considered high cardinality.</td>
</tr>
</tbody>
</table>
Table 27. Global Options and Profile Options  (continued)

<table>
<thead>
<tr>
<th>Option Type</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Size reduction factor</td>
<td>1009</td>
<td>Used when Automatic Sampling is True. The values in a high-cardinality column are reduced by this factor (approximately), and the resulting smaller set is used to determine the column's data type and other results. Also used in the algorithm to determine which values are selected.</td>
</tr>
</tbody>
</table>

**Task Options**

The Task options listed below appear in the indicated steps. When you make a change to any Task option, all steps that uses that option will use the new value the next time they are executed.

Table 28. Task Options

<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binding Condition Discover Correlations</td>
<td>2-column correlation discovery batch size</td>
<td>50000</td>
<td>Number of rows per batch (unit of processing) while performing 2-column correlation discovery.</td>
</tr>
<tr>
<td>Maps Discover Correlations</td>
<td>2-column correlation discovery cardinality difference threshold</td>
<td>0.3</td>
<td>Discovery will only perform 2-column correlation discovery if ((target column cardinality) x (this value)) is equal to or less than the 2-source-column cardinality.</td>
</tr>
<tr>
<td>Reverse Pivot Discover Correlations</td>
<td>2-column correlation discovery for numeric columns only</td>
<td>TRUE</td>
<td>True: 2-column correlation discovery is performed for numeric columns only. False: 2-column correlation discovery is performed for all columns.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>--------------------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Binding Condition Discover Correlations</td>
<td>2-column correlation threshold</td>
<td>0.8</td>
<td>In order to be considered a 2-column correlation, the percentage of correlating 2-source-column values and a target column value, given some binding condition, must meet or exceed this threshold.</td>
</tr>
<tr>
<td>Binding Condition Discover Binding Data</td>
<td>Binding condition hit rate</td>
<td>0.8</td>
<td>A binding condition must match more than this number of rows at either the source or the target. For example, if the source has 100 rows, the target has 10 rows, and the binding condition matches 9 rows in source and 9 in target, the source hit rate is 9/100 or 0.09 and the target hit rate is 9/10 or 0.90. If this value is 0.8, the binding condition is still considered valid because the target hit rate is above the value, even though the source hit rate is not.</td>
</tr>
<tr>
<td>Maps Discover Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maps Discover Binding Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reverse Pivot Discover Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maps Discover Aggregates</td>
<td>Binding condition selectivity threshold</td>
<td>0.9</td>
<td>If either the source or the target column selectivity is at or above this value, Discovery considers the expression a valid binding condition; if both are below this value, Discovery does not consider the expression a valid binding condition.</td>
</tr>
<tr>
<td>Transformations Discover Aggregates</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binding Condition Discover Correlations</td>
<td>Cardinality difference threshold</td>
<td>0.3</td>
<td>Discovery multiplies this value by the cardinality of the target column, and if the result is greater than the cardinality of the source column, Discovery concludes that there is no correlation between the two. If the result is less than or equal to the cardinality of the source column, Discovery continues its correlation discovery.</td>
</tr>
<tr>
<td>Maps Discover Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reverse Pivot Discover Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>PF Keys Generate PF Keys</td>
<td>Column names must match exactly to be a PF key pair</td>
<td>False</td>
<td>Determines whether Discovery requires a match between source and target column names in order to consider two columns a primary-foreign key pair. When False, Discovery considers columns as primary-foreign key pairs based on analysis of their contents, even if the column names do not match. When True, column names must match. If all columns in a table use the same column prefix, Discovery first tries to match column names as they are. If that fails, it then removes the prefixes from all columns and tries to match them without prefixes. For example, consider a table called CUSTOMERS with all columns using a “C_” prefix, and a primary key C_CUSTID. If another table ORDERS uses an “O_” prefix for all its columns and it contains an O_CUSTID column, Discovery discovers a match between CUSTOMERS.C_CUSTID and ORDERS.O_CUSTID.</td>
</tr>
</tbody>
</table>
Table 28. Task Options  (continued)

<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maps Discover Conditional Transformations</td>
<td>Combine filters?</td>
<td>Yes</td>
<td>Enables or disables filter grouping. During filter discovery, Discovery creates simple filters (expressions) for each source column, trying to maximize matched rows while minimizing unfiltered rows. In some cases the quality of the discovered filters may be improved by combining more than one filter, using the AND operator. This option controls whether Discovery uses this grouping feature, or not. However, combining filters is time consuming and does not guarantee improved results. To improve performance, Discovery has a turbo mode that optimizes grouping if no perfect simple filters were discovered. (A perfect filter filters the same rows as the transformation condition, leaving no unfiltered rows.) If there are no perfect filters, turbo mode optimizes grouping by trying to discover only two filters: one with the optimal number of matched rows, and another with the optimal combination of matched and unfiltered rows.</td>
</tr>
<tr>
<td>Maps Discover Conditional Transformations for Sparse Columns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Conditional Transformations for Sparse Columns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>Maps Discover Transformation Data</td>
<td>Complete Transformation discovery</td>
<td>0</td>
<td>Determines whether discovery finds only basic transformations or all possible transformations. • 0 finds only basic transformations, including correlations. • 1 finds every possible transformation, including conditional transformations and data rule discovery, but processing memory requirements and time are longer. • 2 finds transformations and data rules. • 3 finds transformations and conditional transformations.</td>
</tr>
<tr>
<td>Binding Condition Discover Binding Data</td>
<td>Consider one to one conditions only</td>
<td>No</td>
<td>Yes: discovers only one-to-one binding conditions. No: discovers binding conditions of all types.</td>
</tr>
<tr>
<td>Maps Discover Binding Data</td>
<td>Constant character threshold</td>
<td>0.2</td>
<td>The minimum number of occurrences of a character in a column, as a percentage of total unique values in the column, for it to be considered a constant. For example, if this value is 0.8, and if in 1800 out of 2000 unique values (90%) in a particular column the third character in the string is an M, then Discovery identifies the M as a constant.</td>
</tr>
</tbody>
</table>
Table 28. Task Options  (continued)

<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binding Condition Discover Correlations</td>
<td>Correlation character overlap threshold</td>
<td>0.2</td>
<td>The minimum percentage of characters that overlap between the values in a source row and a target row, for Discovery to consider the values a row hit. The sequence of the characters is not considered, only the presence or absence in the string.</td>
</tr>
<tr>
<td>Maps Discover Correlations</td>
<td></td>
<td></td>
<td>This option applies in either direction: source compared to target, or target compared to source. For example, consider a source row containing the value ct23nmp, and corresponding target row containing 3hdk2ppl7885xg. Three of the seven characters in the source (2, 3 and p) are also in the target data, giving an overlap of 3/7, or .43.</td>
</tr>
<tr>
<td>Reverse Pivot Discover Correlations</td>
<td></td>
<td></td>
<td>From the other direction, three of fifteen characters in the target, the same 2, 3 and p, match the source data, giving an overlap of 3/15, or .20. If this option is .45, the row comparison is not counted as a row hit. But, if this option is .40, it is counted as a row hit because it exceeded the option from the target direction, even though it fell below it from the source side.</td>
</tr>
</tbody>
</table>
Table 28. Task Options (continued)

<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binding Condition Discover Correlations</td>
<td>Correlation threshold</td>
<td>0.2</td>
<td>The minimum percentage of rows bound by a specific binding condition, where the value of a source column corresponds to some other value of a target column, in order for Discovery to conclude that a correlation exists between the columns. For example, consider source and target tables containing customer records, both with a column called State. Given a particular binding condition, a customer in the source table has a State of ‘CA’, and the same customer in the target table has a State of ‘California’. Another customer in both tables has a State of ‘NY’ in the source, and a State of ‘New York’ in the target, and so on for other customers. If the percentage of customers who have corresponding State values in both tables meets or exceeds the value set here, Discovery concludes that a correlation exists between the State field in the source and target tables.</td>
</tr>
<tr>
<td>Maps Discover Correlations</td>
<td>Correlation weight coefficient</td>
<td>0.4</td>
<td>Used to calculate the correlation weight for a binding condition. The correlation weight for a column is defined as the correlation hit rate of the best correlation, multiplied by this coefficient. Since accidental correlations are more likely than accidental overlaps, subsets, supersets or equalities, the correlation weight is, by default, lower than either subset or equality weight.</td>
</tr>
<tr>
<td>Reverse Pivot Discover Correlations</td>
<td>Correlation weight coefficient</td>
<td>0.4</td>
<td>Used to calculate the correlation weight for a binding condition. The correlation weight for a column is defined as the correlation hit rate of the best correlation, multiplied by this coefficient. Since accidental correlations are more likely than accidental overlaps, subsets, supersets or equalities, the correlation weight is, by default, lower than either subset or equality weight.</td>
</tr>
<tr>
<td>Transformations Discover Correlations</td>
<td>Correlation weight coefficient</td>
<td>0.4</td>
<td>Used to calculate the correlation weight for a binding condition. The correlation weight for a column is defined as the correlation hit rate of the best correlation, multiplied by this coefficient. Since accidental correlations are more likely than accidental overlaps, subsets, supersets or equalities, the correlation weight is, by default, lower than either subset or equality weight.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------------------------</td>
<td>---------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Data Objects Generate Data Objects</td>
<td>Data Object generation includes attribute tables of reference tables</td>
<td>FALSE</td>
<td>Controls what kinds of tables Discovery includes as related tables when it generates data objects. Depends on what types of tables and data sources are involved in the data object discovery. This option is used only for classification-based data object discovery; it is not used for unclassified data object discovery.</td>
</tr>
<tr>
<td>Data Objects Generate Data Objects</td>
<td>Data Object generation includes Reference tables</td>
<td>FALSE</td>
<td>Controls what kinds of tables Discovery includes as related tables when it generates data objects. Depends on what types of tables and data sources are involved in the data object discovery. This option is used only for classification-based data object discovery; it is not used for unclassified data object discovery.</td>
</tr>
<tr>
<td>Column Analysis</td>
<td>Datatype discovery refresh</td>
<td>true</td>
<td>Determines whether previously-discovered date and number data types will be dropped and rediscovered during restaging. Does not affect actual datatype discovery.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maps Generate Table Maps</td>
<td>Depth of source child tables</td>
<td>2</td>
<td>Specifies the maximum number of levels of child tables on the source side that Discovery analyzes during map discovery. (Child levels are defined by primary-foreign key relationships, the structure of the data object to which the table belongs.) Discovery performs map discovery on the parent plus the number of child source tables defined here, and keeps only the most effective of the maps it discovers. For example, the default value 2 means that Discovery analyzes the relationships of a source table to a target table, and if it finds some potential map (based on the binding conditions it discovers), it proceeds to analyze two levels of that source table's children to see if it can derive a more effective map if one or both levels of child tables are included. Discovery makes a copy of the parent map, and then adds the child tables to it one at a time. If any of these resulting maps are more effective than the map based on the source parent alone, the parent-only map is discarded. If the addition of these child tables does not improve the map, then they are not included in the map. A value of 0 here means that only the specified source tables are mapped, without any of their child tables.</td>
</tr>
<tr>
<td>Maps Generate Table Maps</td>
<td>Depth of target child tables</td>
<td>2</td>
<td>Same as Depth of source child tables, but for target tables rather than source.</td>
</tr>
</tbody>
</table>
Table 28. Task Options  (continued)

<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF Keys Generate PF Keys</td>
<td>Determine primary by matching table-column name when statistics match</td>
<td>True</td>
<td>If Discovery identifies more than one potential primary-foreign key with identical statistical strengths between two tables, it needs some other criterion to judge which one should be marked as primary. When this is True, Discovery looks for any matches between the name of the table and the column. If, for one table, the column name is similar to the table name, and for the other table it is not, Discovery chooses the table with similar name as primary. For example, consider a CUSTOMERS table with CUST_ID column and an ADDRESSES table with a CUST_ID column. If for each customer there is exactly one address record, Discovery does not statistically recognize which column is the primary and which is foreign key. However, when this is True, Discovery recognizes that CUST_ID matches the table name CUSTOMERS better than table name ADDRESSES does, and so designates CUSTOMERS.CUST_ID as the primary key and ADDRESSES.CUST_ID as the foreign key. If this option is set to False, Discovery does not use the column-to-table matching to decide which column is primary and which is a duplicate.</td>
</tr>
<tr>
<td>DS Column Analysis Statistics</td>
<td>Discover column data statistics</td>
<td>TRUE</td>
<td>Determines whether column data-related statistics are refreshed.</td>
</tr>
</tbody>
</table>
### Table 28. Task Options (continued)

<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF Keys Generate PF Keys</td>
<td>Discover Composite Primary- Foreign Keys</td>
<td>FALSE</td>
<td>Determines whether Discovery finds composite PF keys during discovery. Default is False (no discovery). Set this to True to discover composite PF keys.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Discovering composite PF keys is resource-intensive. It is recommended that you select specific tables for discovery instead of running the task on all data sources.</td>
</tr>
<tr>
<td>Maps Discover Transformation Data</td>
<td>Discover Transformations between date/time and numeric types?</td>
<td>No</td>
<td>Controls transformation discovery between numeric and date/time data types. No: Discovery does not search for transformations between these two data types. True: Discovery performs transformation discovery between date/time and numeric data types with scale greater than 0.</td>
</tr>
<tr>
<td>Reverse Pivot Discover Transformation Data</td>
<td>Discover two- column arithmetics</td>
<td>True</td>
<td>True: Discovery discovers two-column arithmetic transformations in addition to single-column transformations. False: Discovery discovers only single-column transformations.</td>
</tr>
<tr>
<td>Transformations Discover Transformation Data</td>
<td>Display only best target matches per column?</td>
<td>Yes</td>
<td>Controls binding condition discovery. When left at Yes, if Discovery discovers multiple alternative column matches for the same column, it selects and displays only the strongest match, based on the hit rate, for each column. This results in faster processing and produces only the strongest alternatives. When set to No, the processing takes longer but Discovery displays all of the alternative matches it discovers, including weaker ones.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>--------------------------------------------------</td>
<td>---------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PF Keys Generate PF Keys</td>
<td>Drop unapproved discovered PF Keys</td>
<td>True</td>
<td>Determines whether to drop all unapproved Discovery-discovered PF Keys when re-running PF Key discovery.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF Keys Generate PF Keys</td>
<td>Enable alternative algorithm to generate PFKeys</td>
<td>True</td>
<td>Can be used with Min primary hit rate to identify column as a foreign key.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>When set to True, the discovery results include PF Keys that are incorrectly identified as foreign-foreign.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global</td>
<td>Enable rounding for determining Transformation</td>
<td>FALSE</td>
<td>Change this to True in order to use Rounding scale for determining Transformation hits.</td>
</tr>
<tr>
<td>Reverse Pivot Discover Transformation Data</td>
<td>Enable rounding for determining Transformation hits</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This setting only affects how the hit rate is calculated and does not affect the actual transformation expression.
<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binding Condition Discover Correlations</td>
<td>Equality threshold</td>
<td>0.2</td>
<td>When the percentage of bound target column rows containing the same value as bound source rows exceeds the value set here, Discovery concludes that a correlation based on an equality expression exists between the two columns.</td>
</tr>
<tr>
<td>Maps Discover Correlations</td>
<td>Filter match threshold</td>
<td>0.8</td>
<td>The minimum percentage of target rows that must match both the filtering condition and the test condition, for the filter to be considered valid. For example, assume target column TC equals source column SC for 100 rows. In 82 of these rows, column FC &lt; 5. In other words, column FC &lt; 5 in 82% (.82) of the rows. If this option is 0.8, Discovery considers FC &lt; 5 a valid filter, because the result in the target column is 0.82.</td>
</tr>
<tr>
<td>Reverse Pivot Discover Correlations</td>
<td>Fixed token count threshold</td>
<td>0.8</td>
<td>When the minimum percentage of rows that have the same number of tokens for a certain column exceeds this value, Discovery considers that column to have a fixed number of tokens. For example, if this value is 0.8, and 85 rows in a column have 2 tokens (for example, last name, first name: Doe, John) and 15 have 3 tokens (for example, last name, first name middle initial Public, John Q.), the column is considered to have a fixed number (2) of tokens.</td>
</tr>
<tr>
<td>Transformations Discover Correlations</td>
<td>Foreign keys in first n child columns identifies relationship as Attribute</td>
<td>3</td>
<td>Based on the table with the foreign key(s). Always used in combination with the threshold Foreign keys not in next n child columns identifies relationship as Attribute.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>---------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| PF Keys Discover  | Foreign keys not in next n child columns identifies relationship as Attribute | 2       | Based on the table with the foreign key(s). Always used in combination with the threshold Foreign keys in first n child columns identifies relationship as Attribute. Discovery classifies a table relationship as Parent-Child only if both of the following conditions are met:  
  • There is exactly one foreign key within the first N columns, counting from the left, where N is the value of Foreign keys in first n child columns identifies relationship as Attribute.  
  • There are no other foreign keys within the next N columns to the right of the foreign key column, where N is the value of this option.  
If either one of these conditions is not met, the relationship is not classified as Parent-Child, and classification discovery continues based on other criteria. (If the first condition is not met because there is no foreign key in the specified number of rows, the second condition is not applicable, since it relies on the location of the first key.) |
<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where Clause Discover Where Clause</td>
<td>Generate additional map with reverse where clause?</td>
<td>Yes</td>
<td>Option used during filter discovery; controls whether Discovery generates a map with a reverse filter. For example, if a binding condition filter discovers a WHERE clause ( C = 5 ) as a valid filter for map ( M ), if this option is set to Yes, it also generates a map identical to map ( M ), except that its Where clause is set to a reverse filter ( C \neq 5 ).</td>
</tr>
<tr>
<td>Reverse Pivot Discover Reverse Pivot Maps</td>
<td>Generate additional maps for reverse pivot columns?</td>
<td>FALSE</td>
<td>True: Discovery rediscovers transformations and correlations for the new reverse pivot map. False: Discovery does not rediscover the expressions, but copies the existing ones from the original map into the new reverse pivot map.</td>
</tr>
<tr>
<td>Reverse Pivot Discover Reverse Pivot</td>
<td>Generate reverse pivot maps</td>
<td>FALSE</td>
<td>True: generates a reverse pivot map during the Discover Pivot Columns task. False: does not generate a reverse pivot map. True: generates the pivot maps.</td>
</tr>
<tr>
<td>Data Objects Generate Data Objects</td>
<td>Generate Unique Data Object</td>
<td>TRUE</td>
<td>True: drop unapproved Data Objects and generate new ones; keep (do not drop or generate) approved Data Objects. False: drop all Data Objects (both approved and unapproved) and generate new ones. This option is important when re-running the Generate Data Objects step.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------</td>
<td>---------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| Maps Generate Table Maps | Map generation mode | CHILDREN | Specifies which mode Discovery uses when generating maps.  
Children: Discovery includes whatever child tables are allowed.  
Direct: the other two options are ignored, and maps are created based only on the chosen target table, or on the selected source and target if you specify individual tables on both sides. No other tables are included in this map, such as lookup tables defined as data rules, and associated source tables that require a join. This may lead to unexpected results. |
<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binding Condition</td>
<td>Match group threshold</td>
<td>0.8</td>
<td>The minimum hit rate that allows Discovery to group two column matches together into a single binding condition. If two column matches are grouped together and the hit rate of the group is at least this percentage of the hit rate of the matches on both source and target sides, then Discovery concludes that they match the same rows, and can be combined into a single binding condition. For example, a column match source.last_name = target.last_name has a 90% hit rate at both source and target, and a match source.first_name = target.first_name has an 85% hit rate at both source and target. When both of these matches are grouped together (source.last_name = target.last_name and source.first_name = target.first_name) they have a hit rate of 79% at the source and 81% at the target. Assuming this option is 0.8, Discovery concludes that these matches belong together in a group.</td>
</tr>
<tr>
<td>Binding Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maps</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discover Binding Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF Keys</td>
<td>Matching # of rows to identify a column match</td>
<td>2</td>
<td>A primary-foreign key match requires more matching rows than the value specified here. Any match with this many matching rows or fewer, is not considered. Because it is likely that at least one value (for example, NULL) will match almost every column, this threshold helps separate out accidental matches from meaningful ones.</td>
</tr>
<tr>
<td>Discover PFKey Matches</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Binding Condition Discover Correlations</td>
<td>Max # of 2-column correlations discovered per target column</td>
<td>100</td>
<td>The maximum number of 2-column correlations that Discovery discovers per target column.</td>
</tr>
<tr>
<td>Maps Discover Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reverse Pivot Discover Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reverse Pivot Discover Reverse Pivot</td>
<td>Max % Transformation hit rate to consider for reverse pivot</td>
<td>0.2</td>
<td>If a target column has a transformation row hit rate above this value, the column is not considered a candidate for a pivot column.</td>
</tr>
<tr>
<td>Binding Condition Discover Correlations</td>
<td>Max 1-column correlation hit rate for 2-column correlation discovery</td>
<td>0.1</td>
<td>Discovery will only perform 2-column correlation discovery if the 2-column correlation hit rate is equal to or greater than ((this value) x (the 1-column correlation hit rate)).</td>
</tr>
<tr>
<td>Maps Discover Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reverse Pivot Discover Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maps Discover Conditional Transformations</td>
<td>Max cardinality for multi-case conditional Transformation discovery</td>
<td>8</td>
<td>If the target column’s cardinality is less than or equal to this value, Discovery will try to discover a CASE statement with a WHEN-THEN clause for each unique target value.</td>
</tr>
<tr>
<td>Transformations Discover Conditional Transformations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF Keys Discover Classifications</td>
<td>Max child tables per attribute table</td>
<td>10</td>
<td>The maximum number of child tables a Child Entity table can have. If a table has more than the number of children specified here, Discovery classifies it as a Root Entity table for Operational schemas and as a Reference Table for Data Warehouse schemas.</td>
</tr>
</tbody>
</table>
Table 28. Task Options (continued)

<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maps Discover Conditional Transformations</td>
<td>Max conditional Transformations discovered</td>
<td>3</td>
<td>The maximum number of non-perfect filters Discovery discovers. When the discovery task finds the number of non-perfect filters specified here, it stops filter discovery. (Perfect filters are not limited by this setting; Discovery lists all perfect filters found, whether they exceed this value or not.)</td>
</tr>
<tr>
<td>Maps Discover Conditional Transformations for Sparse Columns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Conditional Transformations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Conditional Transformations for Sparse Columns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>Where Clause Discover Where Clause</td>
<td>Max number of filter matches to combine</td>
<td>4</td>
<td>Number of times Discovery will combine filter matches to discover WHEN condition. Minimum value is 1. Higher values may affect performance. Old definition when this was “Combine Filters”: Enables or disables filter grouping. Works in combination with the option Use turbo mode for where clause discovery?. The default Yes value can affect performance. Change to No to improve performance. During filter discovery, Discovery creates simple filters (expressions) for each source column, trying to maximize matched rows while minimizing unfiltered rows. In some cases the quality of the discovered filters may be improved by combining more than one filter, using the AND operator. This option controls whether Discovery uses this grouping feature, or not. However, combining filters is time consuming and does not guarantee improved results. To improve performance, Discovery has a turbo mode that optimizes grouping if no perfect simple filters were discovered. (A perfect filter filters the same rows as the transformation condition, leaving no unfiltered rows.) If there are no perfect filters, turbo mode optimizes grouping by trying to discover only two filters: one with the optimal number of matched rows, and another with the optimal combination of matched and unfiltered rows.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>--------------------------------------------------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maps Discover Conditional Transformations</td>
<td>Max number of filters discovered</td>
<td>25</td>
<td>Maximum number of best transformation filter matches added per target column. A value of -1 means there is no limit on the number of filter matches discovered.</td>
</tr>
<tr>
<td>Maps Discover Conditional Transformations for Sparse Columns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Conditional Transformations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Conditional Transformations for Sparse Columns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maps Discover Transformation Data</td>
<td>Max number of unmapped rows allowed</td>
<td>20</td>
<td>Discovery will stop complete transformation discovery (or enhancement of the primary transformation with conditional transformation discovery) if it cannot reduce the number of unmapped rows to below this value.</td>
</tr>
<tr>
<td>Transformations Discover Transformation Data</td>
<td>Max number of unmapped rows allowed</td>
<td>20</td>
<td>Discovery will stop complete transformation discovery (or enhancement of the primary transformation with conditional transformation discovery) if it cannot reduce the number of unmapped rows to below this value.</td>
</tr>
<tr>
<td>Maps Discover Data Rules</td>
<td>Max row count for using data rule inline</td>
<td>5</td>
<td>The maximum number of lines a data rule lookup table can have in order for Discovery to use the table inline (incorporated into SQL queries as a CASE statement). If the number of lines in the table exceeds this value, Discovery does not use the data rule inline. The Inline option on the Data Rule definition pane must be checked in order for this option to be used.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>Column Analysis</td>
<td>Max rows to scan to identify DateTimeString column</td>
<td>500</td>
<td>Used for datatype discovery; specifies a hard limit on the number of records Discovery examines in each column of a table, when searching for date/time formats. This maximum limit applies regardless of whether records are null or not null, or whether date or time format data was found or not. Discovery searches a column for date formats and, regardless of what it finds, stops when it examines the number of records specified here.</td>
</tr>
<tr>
<td>Column Analysis</td>
<td>Max rows to scan to identify DateTimeString date or time column</td>
<td>300</td>
<td>Used for datatype discovery; specifies the number of non-null values Discovery examines when searching for separate date formats or time formats. At the default setting Discovery looks at the first 300 non-null values in a file or table searching for separate date formats or time formats; if it does not find any, concludes that the values in this column are neither date nor time datatypes, and stops looking.</td>
</tr>
<tr>
<td>Column Analysis</td>
<td>Max rows to scan to identify DateTimeString date/time column</td>
<td>75</td>
<td>Used for datatype discovery; specifies the number of non-null values Discovery examines when searching for data in date/time combination format. At the default setting Discovery looks at the first 75 non-null values in a column searching for data in a date/time combination format, and if it does not find any it, concludes that there are no date/time values in the table, and stops looking.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Column Analysis</td>
<td>Max rows to scan to identify DateTimeString when none found</td>
<td>50</td>
<td>Used for datatype discovery; specifies the maximum number of non-null values Discovery examines when searching a column for date formats. At the default setting Discovery looks at the first 50 non-null values in a column searching for date formats; if it does not find any, it concludes that the column is not a date datatype, and moves on to examine the next column.</td>
</tr>
</tbody>
</table>
| Maps Discover Data Rules    | Max source-target cardinality ratio threshold                               | 1.5     | Must be greater than 1. For the source/target pair to be considered for creating a data rule, the following must be true, where \( n \) is this option value: \[
\frac{1}{n} \leq \frac{\text{source cardinality}}{\text{target cardinality}} \leq n
\]
This represents the range between which the difference between the source and target column cardinality is considered a good candidate for a data rule. For example, the default value of 1.5 means that if the (source column cardinality)/(target column cardinality) is lower than 1/1.5 or is greater than 1.5, the correlation is considered accidental and is not accepted during data rule discovery. |
<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maps Discover Conditional Transformations</td>
<td>Max unique values in filter buffer</td>
<td>100</td>
<td>Maximum number of unique values for each column that Discovery caches during filter discovery.</td>
</tr>
<tr>
<td>Maps Discover Conditional Transformations for Sparse Columns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Conditional Transformations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Conditional Transformations for Sparse Columns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Where Clause Discover Where Clause</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maps Discover Conditional Transformations</td>
<td>Max values in IN clause</td>
<td>10</td>
<td>The maximum number of unique values a source column can contain in the rows that match the condition you are trying to filter, in order for the filter to be considered valid.</td>
</tr>
<tr>
<td>Maps Discover Conditional Transformations for Sparse Columns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Conditional Transformations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Conditional Transformations for Sparse Columns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Where Clause Discover Where Clause</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The maximum number of unique values a source column can contain in the rows that match the condition you are trying to filter, in order for the filter to be considered valid.

For example, suppose you are trying to discover a transformation for column TC: TC < SC (a filter for the condition TC = SC). Whenever TC = SC, the source column FC contains one of six unique values: ‘Fargo’, ‘Duluth’, ‘Buffalo’, ‘Syracuse’, ‘Ithaca’ or ‘Poughkeepsie’. When TC != SC, the source column FC does not contain any of those values.

There are six unique values in the source column. If this option is set to 6, Discovery generates the following IN filter:

\[
TC = \text{CASE WHEN } FC \in \{\text{‘Fargo’, ‘Duluth’, ‘Buffalo’, ‘Syracuse’, ‘Ithaca’ or ‘Poughkeepsie’}\} \text{ THEN } SC \text{ ELSE . . .}
\]

If this option is set to 5, Discovery does not generate a filter.
Table 28. Task Options (continued)

<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where Clause Discover</td>
<td>Max where clauses discovered</td>
<td>3</td>
<td>The maximum number of non-perfect filters discovered. When the discovery task finds the number of non-perfect filters specified here, it stops filter discovery. (Perfect filters are not limited by this setting; Discovery lists all perfect filters found, whether they exceed this value or not.)</td>
</tr>
<tr>
<td>Maps Discover Correlations Transformations Discover Correlations</td>
<td>Maximum Cardinality Ratio</td>
<td>20</td>
<td>This value is the maximum cardinality ratio allowed.</td>
</tr>
<tr>
<td>Maps Discover Transformation Data Reverse Pivot Discover Transformation Data Transformations Discover Transformation Data</td>
<td>Maximum number of arithmetic Transformations discovered</td>
<td>-1</td>
<td>The maximum number of arithmetic transformations discovered for the target column. Default is -1, which discovers all arithmetic transformations. If too many arithmetic transformations are discovered, change this value to a positive number to limit the number discovered (for example, 50).</td>
</tr>
<tr>
<td></td>
<td>Maximum number of equalities</td>
<td>10</td>
<td>This value is the maximum number of equalities to keep.</td>
</tr>
<tr>
<td>Maps Discover Correlations Transformations Discover Correlations</td>
<td>Maximum number of equalities</td>
<td>10</td>
<td>This value is the maximum number of equalities to keep.</td>
</tr>
</tbody>
</table>
Table 28. Task Options (continued)

<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformations Discover Conditional</td>
<td>Maximum number of filters to combine</td>
<td>Yes</td>
<td>Enables or disables filter grouping.</td>
</tr>
<tr>
<td>Transformations</td>
<td></td>
<td></td>
<td>During filter discovery, Discovery creates simple filters (expressions) for each source column, trying to maximize matched rows while minimizing unfiltered rows. In some cases the quality of the discovered filters may be improved by combining more than one filter, using the AND operator. This option controls whether Discovery uses this grouping feature, or not. However, combining filters is time consuming and does not guarantee improved results. To improve performance, Discovery has a turbo mode that optimizes grouping if no perfect simple filters were discovered. (A perfect filter filters the same rows as the transformation condition, leaving no unfiltered rows.) If there are no perfect filters, turbo mode optimizes grouping by trying to discover only two filters: one with the optimal number of matched rows, and another with the optimal combination of matched and unfiltered rows.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maps Discover</td>
<td>Maximum number of Transformation functions generated</td>
<td>7</td>
<td>The maximum number of transformations Discovery generates in a map for each column in the target table. Discovery stops transformation discovery for a map as soon as it reaches this number of transformations. Increasing this value allows more (possibly higher quality) transformations to be discovered, but discovery time increases. A setting of 0 blocks Discovery from discovering any transformations for the current map.</td>
</tr>
<tr>
<td>Transformation Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maps Discover</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Data DR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reverse Pivot Discover</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformation Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformation Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Data DR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF Keys Discover</td>
<td>Maximum parent row hit rate to identify Reference relationship</td>
<td>0.8</td>
<td>Distinguishes between Parent-Child and Reference relationships. If the row hit ratio between the column in the parent table and the column in the child table is at or below the value specified here, Discovery classifies the relationship between parent and child as Parent-Child. If it is above this value, the relationship is classified as Reference.</td>
</tr>
<tr>
<td>Classifications</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PF Keys Discover</td>
<td>Min # of matching parent-child table name characters to identify Attribute relationship Change - Parent- Child</td>
<td>7</td>
<td>If the table names of the parent and child match from the first character to more than the number of characters specified here, Discovery may classify the relationship between parent and child as Parent-Child. If a child table has more than one parent, Discovery uses this threshold to decide which parent has a Parent-Child relationship; the other is assumed to have a Reference relationship.</td>
</tr>
<tr>
<td>Classifications</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PF Keys Discover</td>
<td>Min # of parent (dimension) tables to identify a Fact table</td>
<td>5</td>
<td>Used in Data Warehouse schemas only. Specifies the minimum number of parent (dimension) tables a table must have in order to be classified as a fact table. If a table has this many or more dimension tables related to it, Discovery classifies it as a fact table; if it has fewer than this, classification discovery continues based on other criteria.</td>
</tr>
<tr>
<td>Classifications</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maps Discover Data Rules</td>
<td>Min % of source-target Transformation to generate data rule</td>
<td>0.8</td>
<td>The minimum percentage of correctly transformed values that a given transformation needs to produce, for Discovery to conclude that it is valid. If a transformation produces fewer than this many correct row values, it is considered not valid for the current map.</td>
</tr>
<tr>
<td>Transformations Discover</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Rules</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reverse Pivot</td>
<td>Min % source row hits for reverse pivot</td>
<td>0.8</td>
<td>Minimum percentage of row hits on the source column for the column to be included in the reverse pivot.</td>
</tr>
<tr>
<td>Discover Reverse Pivot</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reverse Pivot</td>
<td>Min % target row hits for reverse pivot</td>
<td>0.8</td>
<td>The minimum percentage of row hits on the target column for the column to be considered a pivot column.</td>
</tr>
<tr>
<td>PF Keys Generate PF Keys</td>
<td>Min foreign row hit rate to identify column as foreign key</td>
<td>0.8</td>
<td>The minimum row hit rate the foreign key column must meet to be considered part of a primary-foreign key.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>---------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| PF Keys Generate PF Keys | Min primary hit rate to identify column as a foreign key | 0.0 | Can be used with Enable alternative algorithm to generate PFKeys. When set to a non-zero value, deep filtering is done on column matches to consider them as PF Key candidates. This skips column matches that meet all of the following criteria:  
  - `option_value` > Primary hit rate  
  - Column names do not match  
  - Match has anonymous or incidental values (date values, or less than 2 digits) |
<p>| Joins | Min row hit rate for join | 0.2 | Specifies the minimum row hit rate required for Discovery to consider a given join condition as valid between two tables. When a join is performed on two or more source tables and the result is compared to the target and returns a row hit rate above this level, the join is considered valid. |
| PF Keys Generate PF Keys | Min selectivity value to identify column as primary key | 0.8 | The minimum selectivity value for the primary key column, above which Discovery concludes that the relationship can qualify as a primary-foreign key. |
| PF Keys Discover Classifications | Min unique column selectivity to identify relationship as Attribute | 0.8 | Based on the table with the foreign key. If that table has no children and has more than one foreign key, and if one of the foreign keys is unique, then Discovery checks the selectivity of the unique column against this threshold. If the selectivity is greater than the value specified here, Discovery classifies the relationship as Parent-Child. |</p>
<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Analysis</td>
<td>Minimum % DateTimeString rows to identify DateTimeString column</td>
<td>0.95</td>
<td>Threshold used for datatype discovery. Specifies the minimum percentage of records in a column that must contain a date/time datatype of any format, for the column to be considered a date/time datatype. If this threshold is met, Discovery then applies the dateformat_min_single_format threshold to determine the specific format of this column. If this threshold is not met, Discovery concludes that the column is not a date/time datatype, so the dateformat_min_single_format threshold is irrelevant.</td>
</tr>
<tr>
<td>Maps Discover Transformation Data</td>
<td>Minimum % non-null values used to calculate column length</td>
<td>0.5</td>
<td>The minimum percentage of values in a column that must be of a certain length or longer, for Discovery to consider that to be the length of the column. Discovery takes the largest column length value that meets this value. For example, consider a column with 1600 values, of which 200 are 2 characters long, 1000 are 3 characters long, and 400 are 4 characters long. If this value is 0.8, the column length for transformation discovery purposes is 3, since more than 80% of the values (1400 out of 1600, or .875) are at least 3 characters long. A length of 4 columns is not valid, since it fails this value (400/1600 = .25). The column length of 2 also exceeds this value (1600/1600), but the only the longest valid column length is selected.</td>
</tr>
</tbody>
</table>
Table 28. Task Options (continued)

<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Analysis</td>
<td>Minimum % NumberString rows to identify NumberString column</td>
<td>0.95</td>
<td>The minimum percentage of records in a column that must contain a numeric datatype of any format, for the column to be considered a numeric datatype. If this threshold is met, Discovery then applies the numberformat_min_single_format threshold (see below) to determine the specific format of this column. If this threshold is not met, Discovery concludes that the column is not a numeric datatype, so the numberformat_min_single_format threshold is irrelevant.</td>
</tr>
<tr>
<td></td>
<td>Minimum % of non-null values used to calculate column length</td>
<td>0.5</td>
<td>The minimum percentage of values in a column that must be of a certain length or longer, for Discovery to consider that to be the length of the column. Discovery takes the largest column length value that meets this value. For example, consider a column with 1600 values, of which 200 are 2 characters long, 1000 are 3 characters long, and 400 are 4 characters long. If this value is 0.8, the column length for transformation discovery purposes is 3, since more than 80% of the values (1400 out of 1600, or .875) are at least 3 characters long. A length of 4 columns is not valid, since it fails this value (400/1600 = .25). The column length of 2 also exceeds this value (1600/1600), but the only the longest valid column length is selected.</td>
</tr>
<tr>
<td></td>
<td>Minimum cardinality value for overlaps</td>
<td>2</td>
<td>In order to be considered an overlap, both columns must meet or exceed this value.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Generate Target Matches</td>
<td>Minimum cardinality value for target match</td>
<td>2</td>
<td>Defines the minimum cardinality value any column must have in order to be considered a match of any other column, if the selectivity is too low. If a column has a cardinality below this value, Discovery does not look for a matching column. If either cardinality or selectivity is above its exclude value, Discovery continues considering the column for possible matches.</td>
</tr>
<tr>
<td>PF Keys Discover Classifications</td>
<td>Minimum parent-child row hit rate to identify Reference relationship</td>
<td>0.5</td>
<td>Distinguishes between Parent-Child and Reference relationships. If the row hit ratio between the column in the parent table and the column in the child table is at or above the value specified here, Discovery classifies the relationship between parent and child as Parent-Child. If it is below this value, the relationship is classified as Reference.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>Binding Condition Discover Binding Data</td>
<td>Minimum row hit rate for valid target match</td>
<td>0</td>
<td>Determines if a target match is strong enough to use in binding condition discovery. This value is the minimum percentage of identical values for a column in either source or target tables, for Discovery to use that target match in binding condition discovery. For example, if a source column has 5000 rows, a target column has 2000 rows, and Discovery detects that 1850 rows are identical in both, it calculates the source hit rate to be 1850/5000 = 0.37 and the target hit rate to be 1850/2000 = 0.93. If this option is set to 0.8, the target match qualifies for inclusion in a binding condition because the target hit rate is above the value, even though the source rate is not.</td>
</tr>
<tr>
<td>Overlaps Generate Overlaps</td>
<td>Minimum row hits for overlaps</td>
<td>2</td>
<td>To consider the columns an overlap, the number of row hits must exceed this value. For example, if this value is 2, there must be at least 3 row hits to consider the columns an overlap.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Generate Target Matches | Minimum row hits for target match | 2       | The number of overlapping rows above which Discovery considers source and target columns to be a column match. If source and target columns have only this many overlapping rows or fewer, Discovery concludes that they are so distinct they do not constitute a column match. This also excludes them from consideration as binding conditions, since those are based on column matches.
It is likely that at least one value (NULL, for example) will match for almost every column. This option helps separate out accidental matches from meaningful ones. |
<p>| Binding Condition Discover Binding Data | Minimum selectivity for valid binding condition | 0.9     | If either the source or the target column selectivity is at or above this value, Discovery considers the expression a valid binding condition; if both are below this value, Discovery does not consider the expression a valid binding condition. |
| Binding Condition Discover Binding Data | Minimum selectivity for valid target match | 0.8     | The minimum selectivity value of either the source or the target column, before Discovery concludes that the expression constitutes a binding condition between two tables. If either the source or the target column selectivity is at or above this value, Discovery considers the expression a valid binding condition; if both are below, Discovery does not consider the expression a valid binding condition. |
| Overlaps Generate Overlaps | Minimum selectivity ratio for overlaps | 0.1     | In order to be considered an overlap, both columns must meet or exceed this value. |</p>
<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate Target Matches</td>
<td>Minimum selectivity ratio for target match</td>
<td>0.1</td>
<td>Defines the minimum selectivity value any column must have in order to be considered a match of any other column, if the cardinality is too low. If a column has a selectivity below this value, Discovery does not look for a matching column. If either selectivity or cardinality is above its exclude value, Discovery continues considering the column for possible matches.</td>
</tr>
<tr>
<td>Maps Discover Transformation Data</td>
<td>Minimum Transformation hit rate</td>
<td>0.8</td>
<td>In order for Discovery to generate a map, at least one column in the target table must have a non-constant transformation with a row hit rate above the value specified here. If this condition is not met, the map is not generated.</td>
</tr>
<tr>
<td>Column Analysis</td>
<td>Number discovery max record count</td>
<td>-1</td>
<td>Specifies a hard limit on the number of records Discovery examines in each column of a table, when searching for numeric formats. Options are -1 (all rows), 0 (no rows), or the actual number of rows to scan. This maximum limit applies regardless of whether records are null or not null, or whether date or time format data was found or not. Discovery searches numeric formats and, regardless of what it finds, stops if it has examines the number of records specified here.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Column Analysis</td>
<td>Number format max column width</td>
<td>50</td>
<td>Specifies the maximum length of a string, above which Discovery to ignore it during numeric datatype discovery. By default, if any value is longer than 50 characters, Discovery concludes that it cannot be a numeric datatype, regardless of the other values and types in the same column.</td>
</tr>
<tr>
<td>Column Analysis</td>
<td>Number format stop scan value when found</td>
<td>-1</td>
<td>Specifies the number of non-null values Discovery examines in columns for which number formats have been determined. Options are -1 (all rows), 0 (no rows), or the actual number of rows to scan. This is used when searching for alternative numeric formats in columns already identified as numeric datatypes. Once Discovery finds that a column contains numeric values, it focuses on that column and look for alternative formats, to make sure that a consistent format is used in the entire column. If it does not find any alternative formats after this many non-null values, it does not spend any more time scanning this column.</td>
</tr>
<tr>
<td>Column Analysis</td>
<td>Number format stop scan value when not found</td>
<td>100</td>
<td>Specifies the maximum number of non-null values Discovery examines when searching a column for numeric datatypes. At the default setting Discovery looks at the first 100 non-null values in a column searching for numeric formats; if it does not find any it concludes that the column does not contain numeric datatypes, and stops looking.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------</td>
<td>---------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| Maps Discover Conditional Transformations | Number of consecutive enhancement s of primary Transformation | 1       | Number of times Discovery will try to enhance the primary transformation, when either of the following conditions exist:  
• the transformation is less than 100%  
• more unmapped rows exist |
| Match and Merge Analysis | Number of transitive closure iterations | -1      | -1 (default): Discovery Studio only performs transitive closure if it is necessary. The number of transitive closures depends on the data.  
n (any positive number): Discovery Studio performs no more than the specified number of transitive closures.  
0: Transitive closure is not performed. |
| Binding Condition Discover Correlations | Perform 2- column correlation discovery | FALSE   | True: performs 2-column correlation discovery.  
False: does not perform this discovery. |
| Binding Condition Discover Correlations | Perform extended tracing with 2-column correlation discovery | FALSE   | True: extended tracing is performed during 2-column correlation discovery.  
False: extended tracing is not performed. |
<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maps Discovery</td>
<td>Transformation Data</td>
<td>FALSE</td>
<td>True: If a perfect transformation (100% hit rate) exists for the target column, Discovery uses the perfect match as the primary transformation and does not attempt to discover alternative transformations. False: Discovery attempts to discover alternative transformations even if a perfect transformation was found for the target column.</td>
</tr>
<tr>
<td>DS Column Analysis</td>
<td>Statistics</td>
<td>TRUE</td>
<td>Determines whether data type and format-related statistics are refreshed.</td>
</tr>
<tr>
<td>Transformations</td>
<td>Discover Transformation Statistics</td>
<td>TRUE</td>
<td>True: Refresh statistics only for primary transformations. False: Refreshes statistics for all transformations. This can affect performance.</td>
</tr>
<tr>
<td>Data Objects</td>
<td>Generate Data Objects</td>
<td>TRUE</td>
<td>True = drop unapproved Data Objects and generate new ones; keep (do not drop or generate) approved Data Objects. False = keep (do not drop and generate) all Data Objects, whether approved or unapproved. This option is important when re-running the Generate Data Objects step.</td>
</tr>
<tr>
<td>Global</td>
<td>Rounding scale for determining Transformation hits</td>
<td>8</td>
<td>Only used when Enable rounding for determining Transformation hits is set to True. The scale, or number of digits past the decimal that will be rounded off. For example, a scale of 1 means the number is rounded to n.n. A scale of 2 rounds the number to n.nn; 3 rounds the number to n.nnn, and so on.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------------------------</td>
<td>---------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Binding Condition Discover</td>
<td>Source and target overlap</td>
<td>0.2</td>
<td>Used to compare row content. For example, by default, if more than 80% of the values in column SC overlap with those in column TC, Discovery generates a TC = overlap(SC) correlation.</td>
</tr>
<tr>
<td>Correlations</td>
<td>threshold</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maps Discover Correlations</td>
<td>Subset weight coefficient</td>
<td>0.6</td>
<td>Used to calculate the subset correlation weight, superset correlation weight, and overlap correlation weight. Each of these correlation weights is calculated differently. Normally you should not change these coefficient settings.</td>
</tr>
<tr>
<td>Reverse Pivot Discover</td>
<td>Target column subset threshold</td>
<td>0.2</td>
<td>The minimum required percentage of records in which the target column is a proper subset of a source column. In other words, the minimum percentage where every character in the target column is also in the source column.</td>
</tr>
<tr>
<td>Correlations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover</td>
<td>Target column superset</td>
<td>0.6</td>
<td>The minimum required percentage of rows on the target side where target column is a proper superset of a source column. In other words, every character in the source column is also in the target column.</td>
</tr>
<tr>
<td>Correlations</td>
<td>threshold</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>------------------------------------------------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maps Discover Data Rules</td>
<td>Threshold to discover default expression from data rule lookup table</td>
<td>0.8</td>
<td>When Discovery discovers a data rule, it also tries to discover a default transformation for it. For the default transformation to be valid, it must account for at least this percentage of lookup rows. For example, if the lookup table contains (S-&gt;T):(1A-&gt;A, 2B-&gt;B, 3C-&gt;C, NULL-&gt;' '), the default transformation is substring(source, 2, 1). In other words, by default, the target is the second character of the source value. Once the rows that the default transformation can resolve are removed from the lookup table, it contains NULL-&gt;' '. When used in transformations, the data rule looks up the source value in the lookup table, and, if not found, applies the default transformation. In this example, for source NULL, it returns a blank (' '), but for any other value, it returns the second character.</td>
</tr>
</tbody>
</table>

IBM InfoSphere Discovery User Guide
<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maps Discover</td>
<td>Threshold to use non-unique correlation</td>
<td>0.98</td>
<td>When Discovery performs data rule discovery, it tries to find the best non-unique correlation to build the cross-reference on. However, unless this non-unique correlation is perfect (having either a correlation or a hit rate of 100%), any unique correlation has a better correlation or hit rate. This value helps Discovery decide whether to use the non-unique correlation or a unique one. If the best non-unique correlation has a correlation or hit rate at or above this value, Discovery uses it to build a lookup data rule. Otherwise, it uses a unique correlation if one exists.</td>
</tr>
<tr>
<td>Transformations</td>
<td>Transformation hit threshold</td>
<td>0.2</td>
<td>A transformation must produce at least this percentage of correctly transformed values in order for Discovery to conclude that it is valid. If it does not, the transformation is considered not valid for the current map.</td>
</tr>
<tr>
<td>Treat NULL rows as matches</td>
<td>Yes</td>
<td></td>
<td>Option that controls whether Discovery considers null rows during join discovery, or not. In most databases, comparing NULL columns returns false. For example, A = B is false if both A and B are NULL. When this option is set to Yes, Discovery matches null values in a join. For example, if A and B are both NULL, they should match, and Discovery generates a join either when A = B or when (A is null and B is null).</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------------------------------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maps Discover Correlations</td>
<td>Trim correlations and fuzzy matches</td>
<td>FALSE</td>
<td>Change this to True in the following situations:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• To avoid Out of Memory errors when running discovery for a map containing many columns.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• When discovery yields many matches and you want to limit the results to only the best matches.</td>
</tr>
<tr>
<td>Maps Discover Conditional</td>
<td>Unfiltered match threshold</td>
<td>0.8</td>
<td>The maximum acceptable percentage of unfiltered rows that satisfy the filtering condition but not the test condition, for the filter to be considered valid.</td>
</tr>
<tr>
<td>Transformations for Sparse Columns</td>
<td></td>
<td></td>
<td>For example, suppose this value is 0.8 for a target table with 100 rows. For 80 of them, target column TC = SC1</td>
</tr>
<tr>
<td>Transformations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binding Condition</td>
<td>Use DATE, TIME, or DATETIME types in binding conditions?</td>
<td>Yes</td>
<td>Controls whether Discovery considers column matches involving DATE, DATETIME, and TIME datatype columns as potential components of binding conditions.</td>
</tr>
<tr>
<td>Binding Data</td>
<td></td>
<td></td>
<td>If set to No, Discovery does not look for binding conditions based on column matches involving these datatypes.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>PF Keys Generate PF Keys</td>
<td>Use DATE, TIME, or DATETIME types in composite keys?</td>
<td>Yes</td>
<td>Controls whether Discovery considers column matches involving DATE, DATETIME, and TIME datatype columns as potential components of composite keys. If set to No, Discovery does not look for composite keys based on column matches involving these datatypes.</td>
</tr>
<tr>
<td>Binding Condition Discover Binding Data Maps Discover Binding Data</td>
<td>Use FLOAT, DECIMAL or NUMERIC types in binding conditions?</td>
<td>No</td>
<td>Controls whether Discovery considers column of FLOAT datatype, or of DECIMAL or NUMERIC datatype with Scale &gt; 0 as potential components of binding conditions. If set to No, Discovery does not look for binding conditions based on column matches involving these datatypes.</td>
</tr>
<tr>
<td>PF Keys Generate PF Keys</td>
<td>Use FLOAT, DECIMAL or NUMERIC types in composite keys?</td>
<td>No</td>
<td>Controls whether Discovery considers column of FLOAT datatype, or of DECIMAL or NUMERIC datatype with Scale &gt; 0 as potential components of composite keys. If set to No, Discovery does not look for composite keys based on column matches involving these datatypes.</td>
</tr>
<tr>
<td>PF Keys Generate PF Keys</td>
<td>Use left outer joins for relating tables</td>
<td>False</td>
<td>Determines how Discovery discovers joins during map generation. When this is False, Discovery generates the joins as inner joins. When True, Discovery generates joins as left outer joins.</td>
</tr>
<tr>
<td>Steps Using Option</td>
<td>Option Name</td>
<td>Default</td>
<td>Definition</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------</td>
<td>---------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| Binding Condition Discover Correlations | Use turbo mode for 2-column correlation discovery | TRUE | True: only the first batch of rows is considered during 2-column correlation discovery. This is faster but may not yield the best results.  
False: all batches are considered. This takes longer but discovers the highest number of 2-column correlations. |
| Maps Discover Correlations | | | |
| Reverse Pivot Discover Correlations | | | |
| Transformations Discover Correlations | | | |
| Maps Discover Conditional Transformations | Use turbo mode for conditional Transformation discovery? | Yes | Enables or disables filter grouping turbo mode. |
| Maps Discover Conditional Transformations for Sparse Columns | | | |
| Transformations Discover Conditional Transformations | | | |
| Transformations Discover Conditional Transformations for Sparse Columns | | | |
| Maps Generate Table Maps | Use turbo mode for map discovery | TRUE | Option that enables and disables turbo mode for map generation. When option is on, Discovery does not attempt to map child tables if it cannot create a valid map between the root tables of the source and target data objects.  
If Turbo mode is turned off, Discovery attempts to create maps from a combination of source tables in the source data object for each target table in the target data object |
| Where Clause Discover Where Clause | Use turbo mode for where clause discovery? | Yes | Enables or disables filter grouping turbo mode. |
| Binding Condition Discover Binding Data | Use turbo mode to discover binding conditions | Yes | Reduces discovery time by discovering only the strongest binding conditions. If the first discovery pass does not yield any binding conditions, set this to No and re-run the task. |
| Maps Discover Binding Data | | | |
### Table 28. Task Options (continued)

<table>
<thead>
<tr>
<th>Steps Using Option</th>
<th>Option Name</th>
<th>Default</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maps Discover</td>
<td>Variable token high threshold</td>
<td>0.8</td>
<td>Allows Discovery to determine whether the number of tokens is fixed (the same for all values in a column) or variable (differs among values in a column).</td>
</tr>
<tr>
<td>Transformation Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reverse Pivot Discover Transformation Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Transformation Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Transformation Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Transformations Data DR Default</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maps Discover Transformation Data</td>
<td>Variable token low threshold</td>
<td>0.5</td>
<td>Identifies fixed or variable tokens.</td>
</tr>
<tr>
<td>Maps Discover Transformations Data DR Default</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reverse Pivot Discover Transformation Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Transformation Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Transformation Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformations Discover Transformations Data DR Default</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Option Sets

Discovery uses configurable options that determine the thresholds and other values required to process data. Each Discovery project starts with a default option set containing values that should be adequate in most data discovery situations. The options for each step are shown when you [View Options Window](#) in [The Processing Options Dialog](#) for that step. In that dialog, the options for the particular step are grouped further based on the discovery that occurs in the step.

For example, some of the option groups in the Discover Transformations step are [Discover Aggregates](#), [Discover Conditional Transformations](#), and [Discover Conditional Transformations for Sparse Columns](#), in addition to the [Global Options and Profile Options](#) that are available in every step.
The values in the default option set are valid for many mapping projects. However, adjusting the options can improve discovery results for some types of projects. Correctly modifying options to achieve a particular goal is challenging because you must know not only the data being analyzed, but also the impact of each option on performance and accuracy.

To help you, Discovery includes a number of additional Custom Option Sets. Each custom option set has several values modified to optimize results when working with specific types of projects. As you become familiar with modifying options, you can further customize these option sets, or create your own custom option sets, to obtain more focused results in the types of projects typically performed at your site. Custom option sets can be exported from one project and imported into other Discovery projects.
Using Option Sets

You can use the same option set for each task in the Discovery project, or select a different option set for any particular task execution. You can also re-run a step using a different option set to obtain different results.

About this task

For example, you can use the default option set to run PF Keys discovery. If the results yield a very low number of column matches, you can switch to the Thorough Discovery option set and re-run PF Keys discovery. If that still does not yield matches that you know exist, you can make a copy of the Thorough Discovery option set and manually modify the options further, then use it to re-run the step again.

Continuing the example, for the next step, Data Objects, you could select the new option set or use any other option set. If you are not satisfied with the results of Data Objects discovery, you do not need to create another option set; just select the new option set and change one or more Data Objects values in it, then re-run Data Objects using it.

As another example, you can create an option set named SAP Customer Map Options and fine-tune it for mapping SAP customer data. This option set works because you used your knowledge about the data and your experience using Discovery to adjust the option values to fit with SAP data. You can use this option set for other SAP data mapping projects, and even export and give it to another user who is also mapping SAP data. As the other user becomes more experienced, she can improve the option set to be even more precise, and then share this refined option set with you and other team members.

How do Options Affect Results?

Options can be customized to control Discovery behavior and discovery results in several ways.

Tradeoffs

Some options provide better results or performance in one area at the expense of performance or results in another area. The following types of tradeoffs are managed by options:

Speed vs accuracy:

Some options allow you to trade off speed for accuracy. When optimized for speed (turbo mode), results are delivered faster but the accuracy is somewhat lower. When optimized for accuracy (thorough mode), data is thoroughly analyzed and the most accurate results are obtained, but discovery can take significantly longer.

Normal data vs clean data:

Most data sources have some level of dirty data. When this is encountered during discovery, it is counted as a miss or is ignored. Some options require a certain total number or percentage of hits in order to consider the column as a certain type, use the column in a transformation, and so on. These options by default have lower thresholds so more data will pass. Data sources that are known to be clean allow you to raise these thresholds, which improves the quality of the results and reduces time spent analyzing the data later.
A data set is clean if:

- All data types are accurately identified.
- All columns contain the same data type. For example, in a numeric type column, there are no alpha-numeric strings.
- All PF Keys are perfect, with primary selectivity of 1 and foreign side hit rate of 100%.

Small data set vs large data set:

Extremely large data sets can be made easier to work with by changing options that increase the cache size or batch sizes used by Discovery Studio or the Discovery Engine Service.

Looser or tighter restrictions on discovery:

When results are not satisfactory, you can re-run discovery with higher or lower thresholds to see what is found. Changing these options requires a strong understanding of how Discovery ‘thinks’, and you will become more comfortable making these changes as you continue working with Discovery.

Type of Sensitive Data in the Project

Another factor in customizing the option sets is the type of sensitive data in the project. (Sensitive data discovery and the terms used below are discussed in the IBM InfoSphere Discovery Methodology Guide.)

Sensitive Data Discovery I:

In this scenario, only Column Analysis and Target Matches are performed. Other discovery steps are not relevant to the goal, which is to discover all sensitive data. Data type discovery is run with relaxed thresholds to catch any trace of sensitive data.

Sensitive Data Discovery II:

In this scenario all discovery steps are performed, including creating maps. Hidden or implicit sensitive data discovery is still the goal, so low confidence maps are as important as high confidence maps. Thorough discovery is more important than speed.

Sensitive Data Discovery III (data lineage), ETL Migration, and Interface Modernization:

This is a more traditional scenario, where the goal is to discover clean PF Keys, identify dirty data, process the data quickly, and so on.

Default Option Set

About this task

The default option set will yield good results with most data sets. If the results for a particular task are not what you expected, based on the data in the project, you can select one of the Custom Option Sets and re-run the task. You can also modify the values of individual options and re-run the task.
The default option set cannot be modified or deleted. If you modify the value of one or more options in the default option set, Discovery saves the modified copy as a new custom option set. You can modify values of custom options sets as often as needed, copy them, and delete them. Option sets can also be exported and imported into other projects.

Note: Some options are used in more than one Discovery task. When you modify an option, the new value will be used the next time you run or re-run any task that includes the modified option. This can change existing results. To avoid unexpected results, it is a good practice to run tasks individually after you have started modifying options, and to carefully select the appropriate option set for each task.

Custom Option Sets

This section describes the custom option sets included with Discovery. In each custom option set, the default values in several options have been modified to optimize discovery results in specific types of projects. The names of the custom option sets below refer to concepts discussed in the IBM InfoSphere Discovery Methodology Guide.

Custom option sets are provided for each of the sensitive data discovery phases described in [How do Options Affect Results?]. The options that have been changed are the ones most often modified based on data properties. Becoming familiar with the different results that occur based on these options is a good starting point in learning Discovery.

All changed options and their default values are listed and explained below. Note that the same option can appear in several steps; make sure you are changing the correct option.

Which Custom Option Set to Use

The following table indicates which custom option set is likely to improve results in certain types of projects. In some cases, a separate custom option set is provided for clean data.

If a particular project type is not listed here, use the default option set.

- Sensitive data
  - Phase I: [Sensitive Data Discovery Phase I]
  - Phase II, De-ID
    - Normal data: [Sensitive Data Discovery Phase II]
    - Clean data: [Sensitive Data Discovery Phase II and Clean]
- Fast discovery
  - Normal data: Default option set
  - Clean data: [Turbo Discovery in Clean Data]
- Thorough discovery
  - Normal data: [Thorough Discovery]
  - Clean data: [Thorough Discovery in Clean Data]
- Fewer transformations in a large data set: [High Hit Rates Small Map Size]
- Discover all possible transformations at once: [Complete Transformation]

Chapter 6. Reference 715
Sensitive Data Discovery Phase I

This section describes the changes to the default option set that are necessary to create custom option set for various goals and with different types of data. Each option set is named, but you should choose a name that is meaningful for your particular project.

To discover all sensitive data in a project, we recommend using loose thresholds for data type discovery. For example, if data in a column containing strings are loosely typed as numeric type, then the string values are not staged during data type discovery, and therefore are not considered during target matching.

To compensate for this consequence of loose type discovery, we recommend the following methodology:
1. Create the project and import the source and target tables.
2. Select the Sensitive Data Discovery Phase I option set.
3. Instead of performing Column Analysis (data type discovery) as a first step, Discover Target Matches using the thresholds in this option set.
4. After discovering target matches, Perform Column Analysis using these thresholds.
5. Re-run Target Matches using these thresholds.
6. Combine the target matches found in both discovery steps to form the complete matching report.

The following custom option set improves the discovery of sensitive data in a project where maps are not created. Use your knowledge of the data to adjust these values further as needed.

<table>
<thead>
<tr>
<th>Step and Option Group</th>
<th>Modified Options</th>
<th>Default Value</th>
<th>Modified Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Analysis: Column Analysis</td>
<td>Date format min all formats threshold</td>
<td>0.8</td>
<td>0</td>
<td>Ignores loose typing</td>
</tr>
<tr>
<td>Column Analysis: Column Analysis</td>
<td>Minimum % of NumberString rows to identify NumberString column</td>
<td>0.95</td>
<td>0</td>
<td>Ignores loose typing</td>
</tr>
<tr>
<td>Target Matches: Generate Target Matches</td>
<td>Minimum row hit rate for valid target match</td>
<td>2</td>
<td>0</td>
<td>Identifies every overlap</td>
</tr>
<tr>
<td>Binding Conditions: Discover Binding Data</td>
<td>Minimum selectivity for valid target match</td>
<td>0.1</td>
<td>0</td>
<td>Accepts any selectivity level as a target match</td>
</tr>
<tr>
<td>Target Matches: Generate Target Matches</td>
<td>Minimum cardinality value for target match</td>
<td>2</td>
<td>0</td>
<td>Accepts any cardinality level as a target match</td>
</tr>
</tbody>
</table>

Sensitive Data Discovery Phase II

As described in the IBM InfoSphere Discovery Methodology Guide, the goal of Phase II sensitive data discovery is to identify all values that are derived from known sensitive data. For example, rather than only accepting transformations with a high hit rate, any transformation is acceptable. Lowering the minimum hit rate allows weaker transformations to be discovered.

The trade-off is that Discovery evaluates and allows many more alternative maps, bindings, and transformations, resulting in a longer run time.
The following custom option set improves the discovery of sensitive data in a project where maps will be created. Use your knowledge of the data to adjust these values further as needed.

Table 29. Custom option set to improve the discovery of sensitive data

<table>
<thead>
<tr>
<th>Step and Option Group</th>
<th>Modified Options</th>
<th>Default Value</th>
<th>Modified Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate Maps:</td>
<td>Filter match threshold</td>
<td>0.8</td>
<td>0.3</td>
<td>Evaluates all filter conditions and discards the weaker ones. This change may result in long discovery times, because many filters are discovered and evaluated.</td>
</tr>
<tr>
<td>Discover Conditional Transformations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binding Conditions:</td>
<td>Binding condition hit rate</td>
<td>0.8</td>
<td>0.1</td>
<td>Discovers weak binding conditions</td>
</tr>
<tr>
<td>Discover Binding Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binding Conditions:</td>
<td>2-column correlation threshold</td>
<td>0.8</td>
<td>0.1</td>
<td>Discovers weak 2-column correlations</td>
</tr>
<tr>
<td>Discover Correlations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate Maps:</td>
<td>Minimum transformation hit rate</td>
<td>0.8</td>
<td>0.2</td>
<td>Generates weak maps</td>
</tr>
<tr>
<td>Generate Table Maps</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate Maps:</td>
<td>Transformation row hit threshold</td>
<td>0.2</td>
<td>0.1</td>
<td>Discovers weak transformations</td>
</tr>
<tr>
<td>Generate Table Maps</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sensitive Data Discovery Phase II and Clean

This set of options can be used for analyzing sensitive data that you will create maps for, and that you know to be clean. Hit rates for map-related thresholds are the same as in the Sensitive Data Discovery Phase II option set, but other thresholds related to dirty data have been raised because the data is clean. See How do Options Affect Results?

Because Discovery evaluates and allows many more alternative maps, bindings, and transformations than with the default values, the run time is longer.

The following custom option set improves the discovery of sensitive data in a project where maps are created and the data is known to be very clean. Use your knowledge of the data to adjust these values further as needed.

<table>
<thead>
<tr>
<th>Step and Option Group</th>
<th>Modified Options</th>
<th>Default Value</th>
<th>Modified Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate Maps:</td>
<td>Filter match threshold</td>
<td>0.8</td>
<td>0.3</td>
<td>Evaluates all filter conditions and discards the weaker ones. This change may result in long discovery times, because many filters are discovered and evaluated.</td>
</tr>
<tr>
<td>Discover Conditional Transformations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binding Conditions:</td>
<td>Binding condition hit rate</td>
<td>0.8</td>
<td>0.1</td>
<td>Discovers weak binding conditions</td>
</tr>
<tr>
<td>Discover Binding Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binding Conditions:</td>
<td>2-column correlation threshold</td>
<td>0.8</td>
<td>0.1</td>
<td>Discovers weak 2-column correlations</td>
</tr>
<tr>
<td>Discover Correlations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chapter 6. Reference 717
Generate Maps: Generate Table Maps  
Minimum transformation hit rate  
0.8 0.2 Generates weak maps

Generate Maps: Generate Table Maps  
Transformation row hit threshold  
0.2 0.1 Discovers weak transformations

Column Analysis: Column Analysis  
Date format min all formats threshold  
0.8 1 Does not tolerate dirty data for date type

Column Analysis: Column Analysis  
Minimum % of NumberString rows to identify NumberString column  
0.95 1 Does not tolerate any dirty data for number format

PF Keys: Generate PF Keys  
Min selectivity value to identify column as primary key  
0.8 1 Requires perfect PFkeys

PF Keys: Generate PF Keys  
Min foreign row hit rate to identify column as foreign key  
0.8 1 Requires perfect PFkeys

**Thorough Discovery**
This set of options can be used for a thorough analysis of all data in a project. The default values for many options specify to search a certain number of rows before stopping the search, or to stop the search after identifying a certain number of hits. These changes specify to search all data regardless of the results, with the tradeoff of taking longer to run.

The following custom option set improves the discovery of sensitive data in a project. Use your knowledge of the data to adjust these values as needed.

<table>
<thead>
<tr>
<th>Step and Option Group</th>
<th>Modified Options</th>
<th>Default Value</th>
<th>Modified Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discover Transformations: Discover Conditional Transformations</td>
<td>Use turbo mode for conditional transformation discovery</td>
<td>Yes</td>
<td>No</td>
<td>Turns off turbo discovery</td>
</tr>
<tr>
<td>Discover Transformations: Discover Conditional Transformations</td>
<td>Max conditional transformations discovered</td>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Discover Transformations: Discover Correlations</td>
<td>Use turbo mode for two-column correlation discovery</td>
<td>Yes</td>
<td>No</td>
<td>Turns off turbo discovery</td>
</tr>
<tr>
<td>Discover Transformations: Discover Correlations</td>
<td>Two-column correlation for numeric columns only</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Discover Transformations: Discover Transformation Data</td>
<td>Discover transformations between date/time and numeric types?</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Discover Bindings: Discover Binding Data</td>
<td>Use turbo mode to discover binding conditions</td>
<td>Yes</td>
<td>No</td>
<td>Turns off turbo discovery</td>
</tr>
<tr>
<td>Discover Where Clauses: Discover Where Clause</td>
<td>Use turbo mode for where clause discovery?</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

**Turbo Discovery in Clean Data**
This set of options can be used for analyzing data that you know to be clean. See [How do Options Affect Results?](#)
When using a clean data set, the thresholds can be more stringent. This reduces the number of incidental PF Keys and avoids discovering data types that can cause formatting errors and data value truncation. The trade off is that any dirty data in a column will cause the object (PF Key, transformation, correlation, and so on) to not be discovered.

Table 30. Options for analyzing clean data

<table>
<thead>
<tr>
<th>Step and Option Group</th>
<th>Step and Option Group</th>
<th>Default Value</th>
<th>Modified Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Analysis:</td>
<td>Minimum %</td>
<td>0.95</td>
<td>1</td>
<td>Does not tolerate any dirty data for date type</td>
</tr>
<tr>
<td>Column Analysis:</td>
<td>DateTimeString rows to identify DateTimeString column</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Column Analysis:</td>
<td>Minimum %</td>
<td>0.95</td>
<td>1</td>
<td>Does not tolerate any dirty data for number format</td>
</tr>
<tr>
<td>Column Analysis:</td>
<td>NumberString rows to identify NumberString column</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discover PF Keys:</td>
<td>Minimum selectivity</td>
<td>0.8</td>
<td>1</td>
<td>Requires perfect PF Keys</td>
</tr>
<tr>
<td>Discover PF Keys:</td>
<td>value to identify column as primary key</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discover PF Keys:</td>
<td>Minimum foreign row hit rate to identify column as primary key</td>
<td>0.8</td>
<td>1</td>
<td>Requires perfect PF Keys</td>
</tr>
</tbody>
</table>

The following custom option set improves the discovery of sensitive data in a project where the data is known to be very clean. Use your knowledge of the data to adjust these values further as needed.

**Thorough Discovery in Clean Data**

This set of options can be used for a thorough analysis of all data in a project, where the data is known to be clean. See [How do Options Affect Results?](#).

The default values have been changed to search all data regardless of the results. The options for dirty data tolerance have also been raised, to reduce the number of incidental PF Keys and avoid discovering data types that can cause formatting errors and data value truncation. The trade off is a longer time for discovery to run.

The following custom option set improves thorough data discovery in a project that has very clean data sources. Use your knowledge of the data to adjust these values further as needed.

<table>
<thead>
<tr>
<th>Step and Option Group</th>
<th>Modified Options</th>
<th>Default Value</th>
<th>Modified Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Analysis:</td>
<td>Date format min all formats threshold</td>
<td>0.8</td>
<td>1</td>
<td>Does not tolerate any dirty data for date type</td>
</tr>
<tr>
<td>Step and Option Group</td>
<td>Modified Options</td>
<td>Default Value</td>
<td>Modified Value</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------</td>
<td>---------------</td>
<td>----------------</td>
<td>----------</td>
</tr>
<tr>
<td>Transformations: Discover Correlations</td>
<td>Correlation character overlap threshold</td>
<td>2</td>
<td>.4</td>
<td></td>
</tr>
<tr>
<td>Transformations: Discover Correlations</td>
<td>Correlation threshold</td>
<td>2</td>
<td>.4</td>
<td></td>
</tr>
<tr>
<td>Transformations: Discover Correlations</td>
<td>Equality threshold</td>
<td>2</td>
<td>.4</td>
<td></td>
</tr>
<tr>
<td>Transformations: Discover Correlations</td>
<td>Source and target overlap threshold</td>
<td>2</td>
<td>.4</td>
<td></td>
</tr>
</tbody>
</table>
### Complete Transformation Discovery

By default, Discovery finds basic transformations such as correlations and transformations, but not conditional transformations or data rule discovery. This requires additional processing memory and time, and is not always necessary to achieve a project's goals. You can Re-run Transformations and Generate Data Rules to discover the additional transformations.

A more efficient way to discover every possible transformation in a project is to use this option set. Every possible transformation is discovered in a single pass. Processing memory requirements and time are longer than with normal discovery.

<table>
<thead>
<tr>
<th>Step and Option Group</th>
<th>Modified Options</th>
<th>Default Value</th>
<th>Modified Value</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformations: Discover Transformations</td>
<td>Complete transformation discovery</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

---

Complete Transformation Discovery

By default, Discovery finds basic transformations such as correlations and transformations, but not conditional transformations or data rule discovery. This requires additional processing memory and time, and is not always necessary to achieve a project's goals. You can Re-run Transformations and Generate Data Rules to discover the additional transformations.

A more efficient way to discover every possible transformation in a project is to use this option set. Every possible transformation is discovered in a single pass. Processing memory requirements and time are longer than with normal discovery.
ODBC Driver Data Type Conversion

This appendix describes how Discovery reads and converts data types from the different supported ODBC drivers.

- DB2 Data Types
- Oracle 10g and 9i Data Types
- Microsoft SQL Server Data Types
- Teradata Data Types
- Sybase Data Types

DB2 Data Types

This tables describe how Discovery reads and converts data types from a DB2 ODBC driver. **Native Type** and **Data Type** correspond to columns in The Column Analysis Window.

Data types not listed below are skipped.

Table 31. How Discovery reads and converts data types from a DB2 ODBC driver when staging data source is Oracle 9i or 10g

<table>
<thead>
<tr>
<th>DB2 source data type</th>
<th>Native type</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigint</td>
<td>bigint</td>
<td>numeric</td>
</tr>
<tr>
<td>character(n)</td>
<td>char</td>
<td>varchar</td>
</tr>
<tr>
<td>date</td>
<td>date</td>
<td>date</td>
</tr>
<tr>
<td>decimal(m,n)</td>
<td>decimal</td>
<td>decimal</td>
</tr>
<tr>
<td>double precision</td>
<td>double</td>
<td>float</td>
</tr>
<tr>
<td>float(1..21)</td>
<td>real</td>
<td>float</td>
</tr>
<tr>
<td>float(22..53)</td>
<td>double</td>
<td>float</td>
</tr>
<tr>
<td>integer</td>
<td>integer</td>
<td>integer</td>
</tr>
<tr>
<td>real</td>
<td>real</td>
<td>float</td>
</tr>
<tr>
<td>Small Int</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>time</td>
<td>time</td>
<td>time</td>
</tr>
<tr>
<td>timestamp</td>
<td>timestamp</td>
<td>datetime</td>
</tr>
<tr>
<td>varchar(n)</td>
<td>varchar</td>
<td>varchar</td>
</tr>
</tbody>
</table>

Table 32. How Discovery reads and converts data types from a DB2 ODBC driver when staging data source is DB2

<table>
<thead>
<tr>
<th>DB2 source data type</th>
<th>Native type</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigint</td>
<td>bigint</td>
<td>numeric</td>
</tr>
<tr>
<td>character(n)</td>
<td>char</td>
<td>varchar</td>
</tr>
<tr>
<td>date</td>
<td>date</td>
<td>date</td>
</tr>
<tr>
<td>decimal(m,n)</td>
<td>decimal</td>
<td>decimal</td>
</tr>
<tr>
<td>double precision</td>
<td>double</td>
<td>float</td>
</tr>
<tr>
<td>float(1..21)</td>
<td>float</td>
<td>float</td>
</tr>
<tr>
<td>float(22..53)</td>
<td>float</td>
<td>float</td>
</tr>
<tr>
<td>integer</td>
<td>integer</td>
<td>int</td>
</tr>
</tbody>
</table>
Table 32. How Discovery reads and converts data types from a DB2 ODBC driver when staging data source is DB2 (continued)

<table>
<thead>
<tr>
<th>DB2 source data type</th>
<th>Native type</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>real</td>
<td>real</td>
<td>float</td>
</tr>
<tr>
<td>Small Int</td>
<td>smallint</td>
<td>int</td>
</tr>
<tr>
<td>time</td>
<td>time</td>
<td>time</td>
</tr>
<tr>
<td>timestamp</td>
<td>timestamp</td>
<td>datetime</td>
</tr>
<tr>
<td>varchar(n)</td>
<td>varchar</td>
<td>varchar</td>
</tr>
</tbody>
</table>

Oracle 10g and 9i Data Types

This table describes how Discovery reads and converts data types from an Oracle 9i or 10g ODBC driver. Native Type and Data Type correspond to columns in The Column Analysis Window.

Data types not listed below are skipped. These include LONG RAW, BIT, BINARY, BYTE, VARBYTE, BLOB, VARBINARY, and LONG VARCHAR.

Table 33. How Discovery reads and converts data types from an Oracle 9i or 10g ODBC driver when staging data source is Oracle 9i or 10g

<table>
<thead>
<tr>
<th>Oracle 10g or 9i source data type</th>
<th>Native Type</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Char</td>
<td>char</td>
<td>Varchar</td>
</tr>
<tr>
<td>clob</td>
<td>clob</td>
<td>varchar</td>
</tr>
<tr>
<td>date</td>
<td>Timestamp</td>
<td>Datetime</td>
</tr>
<tr>
<td>Decimal</td>
<td>number</td>
<td>Decimal</td>
</tr>
<tr>
<td>Float</td>
<td>double</td>
<td>float</td>
</tr>
<tr>
<td>Integer</td>
<td>number</td>
<td>Decimal</td>
</tr>
<tr>
<td>nchar</td>
<td>-</td>
<td>Varchar</td>
</tr>
<tr>
<td>Number</td>
<td>number</td>
<td>Decimal</td>
</tr>
<tr>
<td>Numeric</td>
<td>number</td>
<td>Decimal</td>
</tr>
<tr>
<td>nvarchar2</td>
<td>nvarchar</td>
<td>Varchar</td>
</tr>
<tr>
<td>rowid</td>
<td>nchar</td>
<td>Varchar</td>
</tr>
<tr>
<td>Small Int</td>
<td>number</td>
<td>Decimal</td>
</tr>
<tr>
<td>timestamp</td>
<td>Datetime</td>
<td>Datetime</td>
</tr>
<tr>
<td>urowid</td>
<td>Undefined</td>
<td>Varchar</td>
</tr>
<tr>
<td>varchar</td>
<td>Varchar</td>
<td>Varchar</td>
</tr>
<tr>
<td>varchar2</td>
<td>Varchar</td>
<td>Varchar</td>
</tr>
</tbody>
</table>

Table 34. How Discovery reads and converts data types from an Oracle 9i or 10g ODBC driver when staging data source is DB2

<table>
<thead>
<tr>
<th>Oracle 10g or 9i source data type</th>
<th>Native Type</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Char</td>
<td>Varchar</td>
<td>Varchar</td>
</tr>
<tr>
<td>clob</td>
<td>clob</td>
<td>Varchar</td>
</tr>
</tbody>
</table>
### Table 34. How Discovery reads and converts data types from an Oracle 9i or 10g ODBC driver when staging data source is DB2 (continued)

<table>
<thead>
<tr>
<th>Oracle 10g or 9i source data type</th>
<th>Native Type</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>Timestamp</td>
<td>Datetime</td>
</tr>
<tr>
<td>Decimal</td>
<td>Decimal</td>
<td>Decimal</td>
</tr>
<tr>
<td>Float</td>
<td>double</td>
<td>float</td>
</tr>
<tr>
<td>Integer</td>
<td>Decimal</td>
<td>Decimal</td>
</tr>
<tr>
<td>nchar</td>
<td>-</td>
<td>Varchar</td>
</tr>
<tr>
<td>Number</td>
<td>Decimal</td>
<td>Decimal</td>
</tr>
<tr>
<td>Numeric</td>
<td>Decimal</td>
<td>Decimal</td>
</tr>
<tr>
<td>nvarchar2</td>
<td>Varchar</td>
<td>Varchar</td>
</tr>
<tr>
<td>rowid</td>
<td>Varchar</td>
<td>Varchar</td>
</tr>
<tr>
<td>Small Int</td>
<td>Decimal</td>
<td>Decimal</td>
</tr>
<tr>
<td>timestamp</td>
<td>Datetime</td>
<td>Datetime</td>
</tr>
<tr>
<td>urowid</td>
<td>Undefined</td>
<td>Varchar</td>
</tr>
<tr>
<td>varchar</td>
<td>Varchar</td>
<td>Varchar</td>
</tr>
<tr>
<td>varchar2</td>
<td>Varchar</td>
<td>Varchar</td>
</tr>
</tbody>
</table>

**Note:** Case insensitivity is not supported for clob. See [About Case Sensitivity](#).

### Microsoft SQL Server Data Types

This table describes how Discovery reads and converts data types from a MSSQL ODBC driver. **Native Type** and **Data Type** correspond to columns in [The Column Analysis Window](#).

Data types not listed below are skipped.

<table>
<thead>
<tr>
<th>MSSQL Server source data type</th>
<th>Native type for Oracle 9i or 10g staging data source</th>
<th>Data type for Oracle 9i or 10g staging data source</th>
<th>Native type for DB2 staging data source</th>
<th>Data type for DB2 staging data source</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigint</td>
<td>bigint</td>
<td>Numeric</td>
<td>bigint</td>
<td>Numeric</td>
</tr>
<tr>
<td>binary</td>
<td>binary</td>
<td>Varchar</td>
<td>skipped</td>
<td>-</td>
</tr>
<tr>
<td>bit</td>
<td>bit</td>
<td>Integer</td>
<td>bit</td>
<td>integer</td>
</tr>
<tr>
<td>char</td>
<td>char</td>
<td>Varchar</td>
<td>char</td>
<td>Varchar</td>
</tr>
<tr>
<td>datetime</td>
<td>datetime</td>
<td>Datetime</td>
<td>datetime</td>
<td>Datetime</td>
</tr>
<tr>
<td>decimal</td>
<td>decimal</td>
<td>Decimal</td>
<td>decimal</td>
<td>Decimal</td>
</tr>
<tr>
<td>float</td>
<td>float</td>
<td>float</td>
<td>float</td>
<td>float</td>
</tr>
<tr>
<td>int</td>
<td>int</td>
<td>Integer</td>
<td>int</td>
<td>Integer</td>
</tr>
<tr>
<td>money</td>
<td>decimal</td>
<td>Decimal</td>
<td>decimal</td>
<td>Decimal</td>
</tr>
<tr>
<td>nchar</td>
<td>nchar</td>
<td>varchar</td>
<td>nchar</td>
<td>varchar</td>
</tr>
<tr>
<td>ntext</td>
<td>ntext</td>
<td>varchar</td>
<td>ntext</td>
<td>varchar</td>
</tr>
<tr>
<td>numeric</td>
<td>decimal</td>
<td>decimal</td>
<td>decimal</td>
<td>decimal</td>
</tr>
<tr>
<td>nvarchar</td>
<td>nvarchar</td>
<td>Varchar</td>
<td>nvarchar</td>
<td>Varchar</td>
</tr>
</tbody>
</table>
### Teradata Data Types

This table describes how Discovery reads and converts data types from a Teradata ODBC driver. **Native Type** and **Data Type** correspond to columns in The Column Analysis Window.

Data types not listed below are skipped.

<table>
<thead>
<tr>
<th>Teradata ODBC Driver Type</th>
<th>Native type for Oracle 9i or 10g staging data source</th>
<th>Data type for Oracle 9i or 10g staging data source</th>
<th>Native type for DB2 staging data source</th>
<th>Data type for DB2 staging data source</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYTEINT</td>
<td>integer</td>
<td>NUMBER(38)</td>
<td>integer</td>
<td>VARCHAR(3)</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>integer</td>
<td>INTEGER</td>
<td>integer</td>
<td>INTEGER</td>
</tr>
<tr>
<td>INTEGER</td>
<td>Integer</td>
<td>INTEGER</td>
<td>Integer</td>
<td>INTEGER</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>Decimal</td>
<td>NUMBER(18)</td>
<td>Decimal</td>
<td>DECIMAL(18,0)</td>
</tr>
<tr>
<td>FLOAT</td>
<td>Float</td>
<td>FLOAT(126)</td>
<td>Float</td>
<td>DOUBLE</td>
</tr>
<tr>
<td>CHAR</td>
<td>Varchar</td>
<td>VARCHAR2(n BYTE)</td>
<td>Varchar</td>
<td>VARCHAR(n BYTE)</td>
</tr>
<tr>
<td>VARCHAR</td>
<td>Varchar</td>
<td>VARCHAR2(n BYTE)</td>
<td>Varchar</td>
<td>VARCHAR(n BYTE)</td>
</tr>
<tr>
<td>LONG VARCHAR</td>
<td>Varchar</td>
<td>VARCHAR2(4000 BYTE)</td>
<td>Varchar</td>
<td>VARCHAR2(4000 BYTE)</td>
</tr>
<tr>
<td>NUMERIC</td>
<td>Decimal</td>
<td>NUMBER(18)</td>
<td>Decimal</td>
<td>DECIMAL(18,0)</td>
</tr>
<tr>
<td>REAL</td>
<td>Float</td>
<td>FLOAT(126)</td>
<td>Float</td>
<td>DOUBLE</td>
</tr>
<tr>
<td>DATE</td>
<td>Date</td>
<td>VARCHAR2(25 BYTE)</td>
<td>Date</td>
<td>VARCHAR(25)</td>
</tr>
<tr>
<td>TIME</td>
<td>Time</td>
<td>VARCHAR2(25 BYTE)</td>
<td>Time</td>
<td>VARCHAR(25)</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>DateTime</td>
<td>VARCHAR2(25 BYTE)</td>
<td>DateTime</td>
<td>VARCHAR(25)</td>
</tr>
</tbody>
</table>

**Note:** Case insensitivity is not supported for ntext, sql_variant, or text. See About Case Sensitivity.
## Sybase Data Types

This table describes how Discovery reads and converts data types from a Sybase ODBC driver. **Native Type** and **Data Type** correspond to columns in The Column Analysis Window. Data types not listed below are skipped.

<table>
<thead>
<tr>
<th>Sybase ODBC Driver Type</th>
<th>Native type for Oracle 9i or 10g staging data source</th>
<th>Data type for Oracle 9i or 10g staging data source</th>
<th>Native type for DB2 staging data source</th>
<th>Data type for DB2 staging data source</th>
</tr>
</thead>
<tbody>
<tr>
<td>bigint</td>
<td>skipped</td>
<td>-</td>
<td>bigint</td>
<td>Numeric</td>
</tr>
<tr>
<td>binary</td>
<td>skipped</td>
<td>-</td>
<td>binary</td>
<td>skipped</td>
</tr>
<tr>
<td>bit</td>
<td>bit</td>
<td>integer</td>
<td>bit</td>
<td>Integer</td>
</tr>
<tr>
<td>char</td>
<td>char</td>
<td>varchar</td>
<td>char</td>
<td>varchar</td>
</tr>
<tr>
<td>date</td>
<td>date</td>
<td>datetime</td>
<td>date</td>
<td>DateTime</td>
</tr>
<tr>
<td>datetime</td>
<td>datetime</td>
<td>datetime</td>
<td>datetime</td>
<td>DateTime</td>
</tr>
<tr>
<td>decimal</td>
<td>decimal</td>
<td>decimal</td>
<td>decimal</td>
<td>Decimal</td>
</tr>
<tr>
<td>double precision</td>
<td>float</td>
<td>float</td>
<td>skipped</td>
<td>-</td>
</tr>
<tr>
<td>float</td>
<td>float</td>
<td>float</td>
<td>Float</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>int</td>
<td>integer</td>
<td>int</td>
<td>Integer</td>
</tr>
<tr>
<td>longsysname</td>
<td>skipped</td>
<td>-</td>
<td>varchar</td>
<td>varchar</td>
</tr>
<tr>
<td>money</td>
<td>money</td>
<td>decimal</td>
<td>money</td>
<td>Decimal</td>
</tr>
<tr>
<td>nchar</td>
<td>char</td>
<td>varchar</td>
<td>nchar</td>
<td>varchar</td>
</tr>
<tr>
<td>nvarchar</td>
<td>varchar</td>
<td>varchar</td>
<td>varchar</td>
<td>varchar</td>
</tr>
<tr>
<td>numeric</td>
<td>numeric</td>
<td>Numeric</td>
<td>numeric</td>
<td>Numeric</td>
</tr>
<tr>
<td>nvarchar</td>
<td>varchar</td>
<td>varchar</td>
<td>nvarchar</td>
<td>varchar</td>
</tr>
<tr>
<td>real</td>
<td>real</td>
<td>float</td>
<td>real</td>
<td>Float</td>
</tr>
<tr>
<td>smalldatetime</td>
<td>smalldatetime</td>
<td>datetime</td>
<td>smalldatetime</td>
<td>datetime</td>
</tr>
<tr>
<td>smallint</td>
<td>smallint</td>
<td>integer</td>
<td>smallint</td>
<td>Integer</td>
</tr>
<tr>
<td>smallmoney</td>
<td>smallmoney</td>
<td>decimal</td>
<td>smallmoney</td>
<td>Decimal</td>
</tr>
<tr>
<td>time</td>
<td>time</td>
<td>time</td>
<td>time</td>
<td></td>
</tr>
<tr>
<td>tinyint</td>
<td>tinyint</td>
<td>integer</td>
<td>tinyint</td>
<td>Integer</td>
</tr>
<tr>
<td>unichar</td>
<td>unichar</td>
<td>varchar</td>
<td>unichar</td>
<td>varchar</td>
</tr>
<tr>
<td>univarchar</td>
<td>univarchar</td>
<td>varchar</td>
<td>univarchar</td>
<td>varchar</td>
</tr>
<tr>
<td>unsigned bigint</td>
<td>skipped</td>
<td>-</td>
<td>unsigned bigint</td>
<td>Integer</td>
</tr>
<tr>
<td>unsigned int</td>
<td>skipped</td>
<td>-</td>
<td>unsigned int</td>
<td>Integer</td>
</tr>
<tr>
<td>unsigned smallint</td>
<td>skipped</td>
<td>-</td>
<td>unsigned smallint</td>
<td>Integer</td>
</tr>
<tr>
<td>varchar</td>
<td>varchar</td>
<td>varchar</td>
<td>varchar</td>
<td>varchar</td>
</tr>
</tbody>
</table>

### Types of Relationships Discovered

Discovery discovers the following types of relationships in the source and target tables.
Arithmetic Single-Column Transformations

Arithmetic single-column transformations convert a source value to a target value by adding, subtracting, multiplying, or dividing by some constant, or by rounding. For subtraction and division, the constant may be either before or after the column value (for example, either $SC1 - N$ or $N - SC1$).

Arithmetic Two-Column Transformations

Arithmetic two-column transformations derive a target column by adding, subtracting, multiplying, or dividing two source columns. Two-column arithmetic transformations do not use rounding.

Changing Configuration Parameters

This section describes how to change certain configuration parameters after installation, and how to manually start and stop the components.

Exception Manager Configuration

Exception Manager configuration information is described in the IBM InfoSphere Information Analyzer Exception Manager User Guide. Open the Exception Manager dashboard and click the Help link, then click the icon in the menu bar to display online help.

Discovery Engine Service Information

This section contains information about the Discovery Engine Service. More information is in How IBM InfoSphere Discovery Works.

Start the Discovery Engine Service

About this task

The Discovery Engine Service is started automatically after installation, if there were no connection problems. From then on, it runs continually and restarts automatically when the host system restarts.

When needed, use one of the following methods to manually start the Discovery Engine Service:

- From the Start menu, select Start>Programs>IBM InfoSphere>Discovery>Discovery Engine>Start Discovery Engine Service.
- From the Start menu, select Settings>Control Panel>Administrative Tools>Services, and right-click on IBM InfoSphere Discovery Engine Service, then select Start.

In the Windows Task Manager, the process appears as a service under the installation user’s login name, and is titled ExerosDiscoveryEngineService.exe.

Stop the Discovery Engine Service

Normally, you leave the Discovery Engine Service running all the time. It automatically restarts when Windows restarts.
About this task

When needed, use one of the following methods to manually stop the Discovery Engine Service:

- From the Start menu, select Start>Programs>IBM InfoSphere>Discovery>Discovery Engine>Stop Discovery Engine Service.
- From the Start menu, select Settings>Control Panel>Administrative Tools>Services, and right-click on IBM InfoSphere Discovery Engine Service, then select Stop.

Start Discovery Engine Processes

You do not manually start individual Discovery Engine processes. The Discovery Engine Service automatically spawns one or more Discovery Engine processes on a regular basis, as defined in the Discovery Engine Configuration Utility. These appear in the Task Manager as separate ExerosDiscoveryEngine.exe processes under the installation user's login name.

Stop Discovery Engine Processes

About this task

To manually stop a Discovery Engine process while it is processing, stop the Discovery Engine Service. This shuts down all Discovery Engine processes currently running on that host.

If you have set the Save Intermediate Results option to Yes, progress on any partially completed tasks is saved. Otherwise, all work done so far in the task is lost. (See Profile and Global Options in the IBM InfoSphere Discovery User Guide.)

Change the Discovery Engine Service Logon Account

By default, the Discovery installer selects the Local System account as the Discovery Engine Service logon account. However, some environments require that a specific user account be selected. If Discovery Server cannot execute tasks after installation, you need to select a specific user as the Discovery Engine Service logon account.

About this task

The selected user account must have the following permissions:

- Access to the System DSN ODBC connections on the Discovery Engine Service host. All users on the host usually have this privilege.
- At least Read access to all text files in the project, across the network.
- Permission to start the Discovery Engine Service.

The Discovery Engine Service is always started by this user, regardless of who is actually logged in to the host. The user must have permission to run services on the machine. Depending on how you have set up security on the host, all users may have permission to do this, or the user may require local admin rights or some other level of access.

To change the Discovery Engine Service logon account, perform the following actions on the Discovery Engine Service host:

Procedure

1. From the Control Panel, select Administrative Tools>Services.
2. Right-click on **IBM InfoSphere Discovery Engine Service** and select **Properties**.
3. In the **Properties** window, click the **Log On** tab.

4. Click **This Account**.
5. Enter the name, domain, and machine password of an existing Windows user on this machine with the appropriate access privileges. Use the format `<domain>\<username>` for the name.
6. Click **OK** and close the **Properties** dialog.
7. Stop and then re-start the Discovery Engine Service.
Discovery Engine Configuration Utility

About this task

All Discovery Engine Service parameters are configured during installation. The Discovery Engine Configuration Utility allows you to change these settings at any time after installation.

Changes made in the Discovery Engine Configuration Utility apply only to that particular instance of the Discovery Engine Service. If multiple instances of the Discovery Engine Service are installed, modify each one's configuration as appropriate.

**Note:** Changes made in the Discovery Engine Configuration Utility are not applied until you restart the Discovery Engine Service. Also restart Discovery Studio.

**Procedure**

1. Open the Discovery Engine Configuration Utility under `Start>Programs>IBM InfoSphere>Discovery>Discovery Engine>Discovery Engine Config Utility`.
2. In the utility, expand each option group, as shown. The default values are shown.
3. Change the values as necessary. The options are:
   - **Config File**: Directory where the configuration file is saved. Click **Browse** to select a new location.
   - **Start At**: Time of day when the Discovery Engine Service starts spawning Discovery Engine processes.
   - **Stop At**: Time of day when the Discovery Engine Service stops spawning Discovery Engine processes.

**Note**: When the **Start At** and **Stop At** times are exactly the same, the Discovery Engine Service spawns processes continually.
- **Number of Processes**: Maximum number of Discovery Engine processes that this Discovery Engine Service runs at one time. This value does not affect processes spawned by other Discovery Engine Services. More processes allow more tasks to be run concurrently, but tasks may take longer to complete.

- **FlushLogToServerSecs**: The maximum time between updates to the Discovery Engine log. The log may be updated more frequently if many messages are being recorded.

- **MaxMessagesToBuffer**: Maximum number of messages the Discovery Engine process will keep in memory before it writes the messages to the Discovery Engine log.

- **Scheduled Wait Time**: How frequently the Discovery Engine Service checks the number of active Discovery Engine processes.

- **Active Wait Time**: How long the Discovery Engine Service waits to spawn a new Discovery Engine process, after it detects that there are fewer than the maximum **Number of Processes**.

- **Heartbeat Timeout**: Length of time the Discovery Engine Service waits until it considers a Discovery Engine process as finished.

- **ServerPort**: The Discovery Server port, 9090 by default. Do not change this value unless port 9090 is used by another process on the Discovery Server host. See [Change the Discovery Server Port](#).

- **Server**: Network name of the host where Discovery Server is or will be installed. If Discovery Engine Service and Discovery Server are both on this machine, use either the network name or localhost.

- **NamingProviderPort**: Discovery Server naming service port. Default is 8912.

- **ThreadPoolSize**: Determines the number of Java threads for the Discovery Engine. More threads allow more discovery processes to run concurrently, but can also cause out-of-memory problems leading to application failure. Options are:
  - Single: one thread per CPU
  - Max: (default) four threads per CPU
  - DataSet: one thread per dataset in the current project
  - <numeric value>: a specific number of threads per CPU

- **TempProfilingDir**: Directory where Discovery Engine stores its configuration file. See [Change the Discovery Engine Configuration File Location](#).

- **sqltrace**: Whether to include messages about SQL activities in the trace log.

- **trace**: Determines which levels of trace messages are written to the trace log.

- **debug**: Whether or not to create a log for debug messages. This is saved as the Debug Log and is listed in the Activity Viewer.

- **error**: Whether or not to create a log for error messages. This is saved as the Error Log and is listed in the Activity Viewer.

4. Click **Save** and close the Discovery Engine Configuration Utility.

5. Stop and then re-start the Discovery Engine.

### Change the Discovery Engine Configuration File Location

The Discovery Engine stores its configuration file on the top level of the c:\ drive by default. To use a different location:

**Procedure**

1. Open the Discovery Engine Configuration Utility.

2. In the **TempProfilingDir** field, enter the path of an existing drive or directory, such as g:\ or d: emp.
3. Restart Discovery Engine.

**Results**

Discovery Engine creates a new configuration file on the top level of the specified disk or in the specified directory.

**Note:** To revert to the default location, delete the path in the `TempProfilingDir` field and restart Discovery Engine.

The location used for the Discovery Engine configuration file has the following considerations:

- The file can be located on any host in the same domain and network.
- A mapped drive can be specified in the `TempProfilingDir` field, as long as the drive is on a host in the same domain.
- The drive or directory specified in the `TempProfilingDir` field must be shared.
- If the drive or directory is on another host, the host must not require authentication.
- The Discovery Studio user must have permission to read, write, and modify the directory.

**Discovery Server Information**

This section describes the Discovery Server. For more information, see [How IBM InfoSphere Discovery Works](#).

**Start Discovery Server**

**About this task**

The Discovery Server is started automatically after installation, if there were no connection problems. From then on, it runs continually and restarts automatically when the host system restarts.

When needed, use one of the following methods to manually start the Discovery Server:

- From the Start menu, select Start>Programs>IBM InfoSphere>Discovery>Discovery Server>Start Discovery Server Service.
- In the Control Panel, open Administrative Tools>Services, right-click on IBM InfoSphere Discovery Server Service, then select **Start**.

Discovery Server can take up to one minute to start. When it is started, it appears in the Windows Task Manager as a System service named `ExerosDiscoveryServer.exe` (may be shortened to `EXEROS~1.EXE`).

**Stop Discovery Server**

**About this task**

Use one of the following methods to manually stop the Discovery Server:

- From the Start menu, select Start>Programs>IBM InfoSphere>Discovery>Discovery Server>Stop Discovery Server Service.
- In the Control Panel, open Administrative Tools>Services, right-click on IBM InfoSphere Discovery Server Service, then select **Stop**.
**Use a Different Repository**

**About this task**

You can change the repository specified during Discovery installation by using the [Discovery Server Configuration Utility](#).

**Note:** If Exception Manager shares the Discovery repository, you must also use the Exception Manager Configuration Utility to update Exception Manager to use the same new repository. Instructions are in the *IBM InfoSphere Information Analyzer Exception Manager User Guide*.

**Discovery Server Configuration Utility**

**About this task**

All Discovery Server parameters are configured during installation. The Discovery Server Configuration Utility allows you to change these settings at any time after installation.

**Note:** After making changes here, restart Discovery Server and Discovery Studio.

**Procedure**

1. Open the Discovery Server Configuration Utility under Start>Programs>IBM InfoSphere>Discovery>Discovery Server>Discovery Server Config Utility.
2. Change the option values as necessary. Several options are only valid for
   certain database types. The General options are:
   
   • **Database Type**: Type of database used for the repository.
   • **Database Server Name**: Network name or IP address of the repository's
database server.
   • **Database SID** (Oracle) or **Database Name** (DB2): SID of the Oracle instance,
or the name of the DB2 repository database instance.
   • **Port Number**: Port used by the repository database server. Defaults:
     – Oracle: 1521
– DB2: 50000

- **User Name**: User name associated with the JDBC connection.
- **Password**: Password associated with the JDBC user.
- **Connection URL** displays the JDBC connection URL of the repository database, based on the information entered in the previous fields.

The Advanced options are:

- **HeartBeatInterval**: How often, in seconds, Discovery Server checks for a Discovery Studio heartbeat. If Discovery Server does not find a heartbeat, it maintains the lock on the open project for the length of time specified in **UISessionTimeout**, and then unlocks the project. This allows Discovery Studio to execute long processing tasks on a project without Discovery Server accidentally unlocking the project and allowing another user to edit it.
- **Heartbeat Timeout Factor**
- **CancelGracePeriod**
- **DebugMode**: Logs messages that may be useful to a support engineer in a debug log file.
- **UISessionTimeout**: How long, in seconds, Discovery Server keeps an open project locked after Discovery Studio misses a heartbeat. After this length of time, Discovery Server unlocks the project.
- **MaxRetries**: Maximum number of times a Discovery Engine process checks the Discovery Server queue again for a task, after it has already checked once and not found a task. After this number of retries, the Discovery Engine process ends.
- **HeartbeatInterval**

3. Click **Save**, then **Close**.

**Change the Discovery Server Port**

Discovery Server requires a server port for its JBoss component (9090 by default), and Discovery Engine Service needs to know this port number. If port 9090 is specified during Discovery Engine Service installation but the port is already in use by another application, either change the other application's JBoss port or change the Discovery Server JBoss port.

**About this task**

Instructions for changing the Discovery Server JBoss port are below.

To change the default JBoss port for Discovery Server, perform the following procedure on each Discovery Engine Service host that points to this Discovery Server:

**Procedure**
1. Open the [Discovery Engine Configuration Utility](#).
2. Change the **ServerPort** value to the new port.
3. Save and close the Discovery Engine Configuration Utility.
4. Open the following file in a text editor:
   
   `<install_dir>\Discovery Server\server\exeros\deploy\jboss-web.deployer\server.xml`

5. Find the line shown below and change the value from 9090 to the new port:
   
   ```xml
   Connector port="9090" address="${jboss.bind.address}
   ```

6. Save and close the file.
**Discovery Studio Information**

Before starting Discovery Studio, make sure that both the Discovery Engine Service and Discovery Server are running.

**Start Discovery Studio**

**About this task**

To start Discovery Studio, select Start>Programs>IBM InfoSphere>Discovery>Discovery Studio.

Each time you open Discovery Studio, you are asked for the name of the Discovery Server host and port. If you have several Discovery Servers installed at your site, you can select which Discovery Server to use for each Discovery Studio session. You can also connect to a different Discovery Server while you are working in Discovery Studio.

- **Server** is the name of the host where Discovery Server is installed. If Discovery Server is on the same host as Discovery Studio, leave the default value, localhost (or the network name of the host).
- **Port number** is usually left at 9090. Do not change the port unless you know it was changed from the default 9090 when Discovery Server was installed.
- **Automatically connect to this server** tells Discovery Studio to reconnect to this Discovery Server every time you start Discovery Studio, without displaying this dialog. To redisplay this dialog later and select a different Discovery Server, start Discovery Studio and click **Server Connect** on the Home page.

If you do not check this option, the **Server Connection** dialog will appear each time you start Discovery Studio and contain the last values you entered.

![Server Connection dialog]

**Stop Discovery Studio**

**About this task**

To exit Discovery Studio, select **File>Exit**. You are asked to save any unsaved changes, and the application closes. The Discovery Engine Service and Discovery Server continue running.
Database Type Considerations

This section contains database-specific considerations that are important to know when using Discovery with various database vendors. Unless otherwise specified, the items listed below apply to all versions of the database type.

Also see "Troubleshooting" and ODBC Driver Data Type Conversion.

Database Errors

Error During Transformation Discovery

Due to an Oracle bug, an ORA-600 error may appear during transformation discovery. See Troubleshooting Map Generation.

Join Error in Wide Data Sets

When joining wide source or target data of any type (Oracle, DB2, text files, etc.), and the staging data source is Oracle, or the project contains at least one Oracle source or target table, the following error may occur during discovery:

ODBC Error Message [HY000:1445]: [Oracle][ODBC][Ora]ORA-01445: cannot select ROWID from, or sample, a join view without a key-preserved table.

If this happens, set the Enable Oracle Join option to True. Discovery will use Oracle-specific joins for all queries, which will allow successful discovery.

Data Types

Numberstring Data Type in Text Files

When importing a text file into an Oracle staging data source, the maximum length of a Numberstring data type is 38. The maximum length when using a DB2 staging data source is 31. See Column Analysis Data Grid.

Oracle URITYPE and UROWID Data Types

Due to a bug in the Oracle ODBC driver, tables with URITYPE and UROWID data types must be imported from an Oracle 10G R2 server. See General Troubleshooting.

ODBC Drivers

DateTime Format

If the ODBC driver for a source or target table uses a different DateTime format than the format discovered or read from metadata during Column Analysis, the values in the Column Analysis Min and Max columns are incorrect. See Change the Date, DateTime, or DateTimeString Format.

Sybase ODBC Driver Options

When using a Sybase ODBC driver, the Use Quoted Identifier and Enable Dynamic Prepare options must be active (checked). See Troubleshooting Data Sets.
Sort Order

Because different database types can use different sort orders, unexpected
discovery results may occur when a source table is a different database type than
the staging data source. Make sure all source database types use the same sorting
order as the staging data source. See Troubleshooting Different Database Types.

Thresholds

The default value for the Max number of columns for aggregation discovery and
cardinality queries threshold depends on the staging data source database type.

Truncation

During transformation discovery, Discovery calculates numbers to the maximum
decimal places. The database used for the staging data source may automatically
truncate or round off these results. See Level of Decimal Precision.

When importing tables with long numbers into a DB2 staging data source, you can
either truncate the numbers to a precision of 31 places or treat the entire column as
VARCHAR. See Handling long database table numbers when DB2 is staging
database.
Chapter 7. Troubleshooting

This section describes common problems encountered when using Discovery Studio.

- Configuration Questions
- Problems During Startup or Running
- Discovery Studio Login Errors
- Discovery Studio Operation Errors

## Configuration Questions

The table provides answers to general Discovery configuration questions.

### Table 35. Discovery configuration

<table>
<thead>
<tr>
<th>Question</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>How do I configure which Discovery Server is monitored by a Discovery Engine Service?</td>
<td>In the Discovery Engine Configuration Utility, enter the host name and port of the Discovery Server that you want this Discovery Engine Service to be associated with. (This information is in the Discovery Server Configuration Utility.)</td>
</tr>
<tr>
<td>How do I configure which Discovery Server my Discovery Studio is connected to?</td>
<td>When you open Discovery Studio, you specify a Discovery Server host and port in the login screen. You can also connect to a different Discovery Server while you are working in Discovery Studio.</td>
</tr>
<tr>
<td>How do I configure which Discovery Engine Service will run my Discovery Studio tasks?</td>
<td>In Discovery Studio, you connect to a specific Discovery Server. One or more Discovery Engine Services are associated with that Discovery Server. One of these Discovery Engine Services will create a Discovery Engine process to execute your task; you cannot specify which one.</td>
</tr>
<tr>
<td>How do I use a different repository with the Discovery Server?</td>
<td>Shut down Discovery Server. In the Discovery Server Configuration Utility, change the repository database settings, then re-Start Discovery Server. You will have access to the projects in the new repository, but not to any projects in the old repository.</td>
</tr>
<tr>
<td>How do I run more than one Discovery Engine Service on the same host?</td>
<td>You can run only one Discovery Engine Service on a host.</td>
</tr>
<tr>
<td>How do I configure a Discovery Engine Service to run more than one Discovery Engine at a time?</td>
<td>In the Discovery Engine Configuration Utility, change the Number of Processes option to 2 or more. This increases the number of simultaneous Discovery Engine processes created by the Discovery Engine Service. However, tasks may execute more slowly or run out of memory if several memory-intensive tasks are being executed at the same time.</td>
</tr>
<tr>
<td>How do I edit the configuration file for the Discovery Server or Discovery Engine Service?</td>
<td>See Discovery Engine Configuration Utility and Discovery Server Configuration Utility.</td>
</tr>
<tr>
<td>How do I change the Exception Manager repository?</td>
<td>Use the Exception Manager Configuration Utility. See the IBM InfoSphere Information Analyzer Exception Manager User Guide.</td>
</tr>
</tbody>
</table>
### Table 35. Discovery configuration (continued)

<table>
<thead>
<tr>
<th>Question</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>How do I resolve a port conflict?</td>
<td>This occurs when Discovery Server is configured to use a port that another application is using, and the Discovery Engine Service cannot connect to Discovery Server. You can use the netstat command to find out which process is bound to the port and then kill it. Contact Customer Support for instructions for changing the Discovery configuration files to use different ports.</td>
</tr>
</tbody>
</table>

### Problems During Startup or Running

This table describes problems with starting or running Discovery Server, Discovery Engine Service, or Discovery Studio, and how to correct them.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Suggested Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Discovery Engine Service did not start after installation.</td>
<td>The Discovery Server host or port configured during installation was not correct. Open the <a href="#">Discovery Engine Configuration Utility</a> and make sure the host and port are correct, then <a href="#">Start the Discovery Engine Service</a>.</td>
</tr>
<tr>
<td>The Discovery Server did not start after installation.</td>
<td>The JDBC connection configured during installation was not correct, and the installer could not start the Discovery Engine Service. Open the <a href="#">Discovery Server Configuration Utility</a> and make sure the connection information is correct, then <a href="#">Start Discovery Server</a>. Port conflicts may keep the Discovery Server from starting. Contact Customer Support.</td>
</tr>
<tr>
<td>When I start Discovery Studio and try to connect to the server, an error message states: Network error between Discovery Studio and Discovery Server: Unable to connect to the remote server.</td>
<td>Make sure the correct Discovery Server <a href="#">Server</a> and <a href="#">Port Number</a> are selected in the Server Connection dialog when you start Discovery Studio. Verify that the Discovery Server service is running. If Discovery Server is on a different host, make sure the host is running and there are no firewall or other access issues. Verify that the user name, password, and JDBC connection are correct (see <a href="#">Discovery Server Configuration Utility</a>).</td>
</tr>
<tr>
<td>Discovery Studio cannot connect to the IBM DB2 Express Edition repository.</td>
<td>DB2 may have timed out before starting. Verify that DB2 is running by opening <a href="#">Start &gt; Control Panel &gt; Administrative Tools &gt; Services</a> and checking the status of the DB2 - DB2COPY1 - DB2 service. Start the service if it is not already running. If it is running, stopping and then restarting the service usually resolves this issue.</td>
</tr>
<tr>
<td>I need to connect to a different Discovery Server, but the Server Connection dialog doesn't appear when I start Discovery Studio.</td>
<td>In a previous login, you check the option <a href="#">Automatically connect to this server</a> in the Server Connection dialog. To display the dialog again, go to the Home page and click <a href="#">Server Connect</a>, then uncheck the option. (see <a href="#">Start Discovery Server</a>)</td>
</tr>
<tr>
<td>Discovery Engine is on a different network drive, and none of my Discovery Studio tasks are executing.</td>
<td>Discovery Engine is installed by default with a default user of Local System. This user may not have network access. <a href="#">Change the Discovery Engine Service Logon Account</a> to a user with network access.</td>
</tr>
</tbody>
</table>
Problem | Suggested Solution
--- | ---
My Discovery Server is installed on a different host; how do I tell if it is running? | Specify the Discovery Server host and port in the Discovery Studio Server Connection dialog. If Discovery Studio successfully starts, that Discovery Server is running. If it is not running, you will get a connection error message. (see [Start Discovery Server](#))
I tried to open the Release Notes (or any online help file) but got a message that the file cannot be found or the shortcut has been changed. | The documentation file is not in its expected location in the installation directory. It may have been moved after installation, or not installed. Re-install Discovery, or fix the shortcut if it offers the option.
I’m sure I fixed my problem according to instructions, but the problem persists. | If you changed any Discovery Server or Discovery Engine Service configuration settings, **Stop Discovery Server** or **Stop the Discovery Engine Service**, then restart it.
You may need to **Stop the Discovery Engine Service** using the Task Manager.

### Discovery Studio Login Errors

This table describes how to correct various Discovery Studio login errors.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Suggested Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>An error message states you cannot connect to the repository server, or no projects appear in the Home tab.</td>
<td>The repository database connection is not correctly configured. Open the <a href="#">Discovery Server Configuration Utility</a> and verify that the <strong>UserName</strong> and <strong>Password</strong> values are correct.</td>
</tr>
<tr>
<td>“Database init failed; Nested exception is: JDBC connection could not be established with connection URL.”</td>
<td>The JDBC URL for the repository database is not correct. Open the <a href="#">Discovery Server Configuration Utility</a> and verify that the connection values are correct.</td>
</tr>
<tr>
<td>“Cannot commit JDBC connection for repo database.”</td>
<td>Discovery Server has lost its connection to the repository database because the repository database server has shut down or there are network or other problems external to Discovery. Contact your network administrator.</td>
</tr>
<tr>
<td>“The underlying connection was closed. Unable to connect to the remote server.”</td>
<td>Discovery Server is not running. Exit Discovery Studio, <strong>Stop Discovery Server</strong>, and then restart it, then <strong>Start Discovery Studio</strong> again. The host or port specified in the Server Connection dialog is incorrect. Verify these values. (see <a href="#">Start Discovery Studio</a>)</td>
</tr>
<tr>
<td>“Version Mismatch: Client version &lt;4.x&gt; cannot access the server version &lt;4.y&gt;.”</td>
<td>The Discovery Server and Discovery Studio versions are different. Upgrade both components to the latest version. (See the IBM InfoSphere Discovery Release Notes.)</td>
</tr>
</tbody>
</table>

### Discovery Studio Operation Errors

This section describes errors that can occur while working in Discovery Studio.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Suggested Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Troubleshooting</td>
<td></td>
</tr>
<tr>
<td>Problem</td>
<td>Suggested Solution</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>While working in Discovery Studio, you see error messages similar to</td>
<td>The maximum number of connections to one of the databases has been reached. Make sure all inactive connections are released, or raise the maximum number of connections.</td>
</tr>
<tr>
<td>the following:</td>
<td></td>
</tr>
<tr>
<td>Server: ORA-00018: maximum number of sessions exceeded</td>
<td></td>
</tr>
<tr>
<td>All of the projects are missing from the Home tab.</td>
<td>Make sure you are viewing the correct tab in the Home window (Transformation Discovery or SDD Discovery). Discovery Studio is not connected to Discovery Server. Make sure Discovery Server is running and Discovery Studio is connected.</td>
</tr>
<tr>
<td>Failed to list project error</td>
<td>This can occur when multiple Java processes are blocking the Discovery Server ports, keeping the JBoss service from connecting to Discovery Server.</td>
</tr>
<tr>
<td>Failed to list project: JDBC connection could not be established with</td>
<td>Stop Discovery Server. Display the Windows Task Manager and manually end all running Java processes, then restart Discovery Server.</td>
</tr>
<tr>
<td>connection URL &lt;Discovery_Server_JDBC_URL&gt; and user name &lt;user_name&gt;</td>
<td></td>
</tr>
<tr>
<td>Grids or diagram layouts are incorrect: rows or columns are missing or</td>
<td>On rare occasions, a filter applied to a grid in one tab becomes erroneously applied to all grids in all tabs. Or, a diagram layout applied to one diagram is applied to all diagrams.</td>
</tr>
<tr>
<td>columns are not saved.</td>
<td>To clear this problem, close the project and select Tools &gt; Clear All User Settings. This deletes the saved filters, grid layouts, and diagram layouts for the current Windows user.</td>
</tr>
<tr>
<td>Troubleshooting Data Sets</td>
<td></td>
</tr>
<tr>
<td>When importing text files, text delimiters that appear at the</td>
<td>When importing text files, carefully preview the data in the Column Analysis window and make changes as needed.</td>
</tr>
<tr>
<td>beginning of a text record are not converted correctly.</td>
<td></td>
</tr>
<tr>
<td>When using a Sybase source or target database, errors occur: tables</td>
<td>Make sure the Sybase ODBC driver has the Use Quoted Identifier and Enable Dynamic Prepare options checked, in addition to the default options (Set ANSI Null and Cumulative Record Count).</td>
</tr>
<tr>
<td>do not import correctly, or discovery does not work.</td>
<td></td>
</tr>
<tr>
<td>URITYPE and UROWID errors (Oracle)</td>
<td>Due to a bug in the Oracle ODBC driver, tables with URITYPE and UROWID data types must be imported from an Oracle 10G R2 server.</td>
</tr>
<tr>
<td>Oracle data types URITYPE and UROWID are not read correctly in</td>
<td></td>
</tr>
<tr>
<td>Discovery.</td>
<td></td>
</tr>
<tr>
<td>Troubleshooting Map Generation</td>
<td></td>
</tr>
<tr>
<td>No maps were created.</td>
<td>See <a href="#">If No Maps Were Created</a>.</td>
</tr>
<tr>
<td>When you generate maps, ODBC ‘invalid number’ errors or internal</td>
<td>This type of problem can occur if you reuse an existing staging data source. When you reuse a staging data source, make sure it is clean before you start.</td>
</tr>
<tr>
<td>consistency errors appear in the log. The task may or may not</td>
<td></td>
</tr>
<tr>
<td>complete successfully.</td>
<td></td>
</tr>
<tr>
<td>During transformation discovery for maps, Discovery Server displays an</td>
<td>This is an Oracle bug for a low-level, unexpected condition. Upgrade to the latest release. Patches are available for the following releases on Windows 2003 Server:</td>
</tr>
<tr>
<td>Oracle ORA-600 error and starts mapping the next column.</td>
<td>• For Oracle 9.2.0.1, install patch 2264056.</td>
</tr>
<tr>
<td></td>
<td>• For Oracle 10.1.0.2, install patch 2982983.</td>
</tr>
<tr>
<td></td>
<td>• Oracle 9.2.0.8 and 10.2 for Windows 2003 Server already contain the patch.</td>
</tr>
<tr>
<td>Problem</td>
<td>Suggested Solution</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>(Applies only when staging data source is Oracle version 9.2 or above, or 10.x. Not seen in Oracle 11.) A Discovery task contains failures involving joins across wide tables.</td>
<td>Change the <a href="#">Enable Oracle Join</a> option to True. If errors still occur, install the appropriate Oracle patch. Contact IBM Customer Support for more information.</td>
</tr>
</tbody>
</table>

## Troubleshooting Project Imports and Exports

**Importing a project fails with a message similar to the following:**

Failed to Import Project: `<longfilename.xml>` is missing for Object `<longobjectname>`

One possible cause is that the exported files were saved into or later copied into a directory with a long file path, causing the absolute path of the export files to exceed the Windows limit of 255 characters. As a result, Windows saved the files with truncated filenames. During import, Discovery looks for the original filenames stored in the export catalog and cannot find them, so the import fails.

When saving or copying exported files, make sure Windows does not truncate the filenames.

**When exporting a project to a directory that already contains files, Discovery deletes all files in the directory and replaces them with the project files.**

This is the normal behavior. Be very careful when using the File>Export command. Make sure the specified directory is appropriate, and is either empty or contains only old project files.

## Troubleshooting Task Execution

**Discovery Server cannot execute tasks.**

Make sure the Discovery Engine Service logon account has all required permissions. See [Change the Discovery Engine Service Logon Account](#).

**While executing a task, errors similar to the following appear in the Error List:**

- The transaction log for the database is full
- Unable to extend to temp space

Increase the size of the transaction log in the database. If you cannot increase the size of the transaction log, define a Sample Set that is smaller than the original data set and repeat the task.

**The Discovery Studio or Discovery Engine appears to hang (no progress after a long time) when you perform any of the following actions:**

- Binding condition discovery
- Display Hits or Misses for join or binding conditions
- Display Source, Target, or Query Results in the Preview pane
- Focus

Discovery Studio or the Discovery Engine may be performing complex transformations. This happens when the source and target columns used in the binding condition, or the source columns used in the join condition, are of different data types. Specifically, one column is VARCHAR and the other column is NUMERIC.

In the Data Types window, change the datatype of one of the columns to match the other column (either both VARCHAR or both NUMERIC). Discovery will not need to perform the transformations and processing will be much quicker.
<table>
<thead>
<tr>
<th>Problem</th>
<th>Suggested Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Discovery component hangs during certain activities:</td>
<td>This happens when the Discovery Server host has a network adapter that cannot accept incoming connection requests (Oracle requires such a network adapter). Discovery components that use JBOSS are trying to connect to Discovery Server's IP address and are failing.</td>
</tr>
<tr>
<td>- Discovery Studio hangs when you click Tools&gt;Validation Jobs.</td>
<td>By default, JBOSS uses all adapters on a host. Two possible solutions are:</td>
</tr>
<tr>
<td>- Discovery Engine hangs during task execution.</td>
<td>- Remove the adapter from the Discovery Studio host.</td>
</tr>
<tr>
<td>- Exception Manager Engine hangs during task execution or does not pick up a job.</td>
<td>- On the Discovery Server host, configure JBOSS to use an adapter that can accept incoming connection requests. See the JBOSS documentation for instructions.</td>
</tr>
<tr>
<td>You try to discover primary-foreign keys, but the task fails with an Insufficient Privileges error message.</td>
<td>Make sure the ODBC connection has privileges on the staging schema.</td>
</tr>
<tr>
<td>When you discover PF Keys, the Hit Rates and Selectivity are either 0 or 1 for all columns.</td>
<td>The ODBC driver you used to define the tables in the data source does not support array read. Use an ODBC driver that does support this.</td>
</tr>
<tr>
<td>(Oracle 10.2 repository and staging data source only)</td>
<td>The Oracle host may not have Remote Desktop Users enabled. On the Oracle host, perform the following actions:</td>
</tr>
<tr>
<td>When running a task, the following error occurs:</td>
<td>1. In the Control Panel, select Administrative Tools &gt; Local Security Policy &gt; User Rights Assignment &gt; Create global objects.</td>
</tr>
<tr>
<td>Failed to connect to datasource conn3 ODBCError: ODBC Error Message [IM003:160]: Specified driver could not be loaded due to system error 1114 (Oracle in OraClient10g_home1).</td>
<td>2. In the Create global objects Properties dialog, click Add User or Group... .</td>
</tr>
<tr>
<td></td>
<td>3. In the Select Users or Groups dialog, click Object Types... .</td>
</tr>
<tr>
<td></td>
<td>4. Make sure the Groups object type is checked, then click OK to close the dialog.</td>
</tr>
<tr>
<td></td>
<td>5. In the Select Users or Groups dialog, click Locations .</td>
</tr>
<tr>
<td></td>
<td>6. In the Locations dialog, highlight the name (not the domain) of the Discovery Server host, then click OK .</td>
</tr>
<tr>
<td></td>
<td>7. In the Select Users or Groups dialog, type Remote Desktop Users in the Enter the object names to select field, then click OK .</td>
</tr>
<tr>
<td></td>
<td>8. Click OK to close the Create global objects Properties dialog.</td>
</tr>
<tr>
<td></td>
<td>If the error still occurs, reboot the Oracle host.</td>
</tr>
<tr>
<td>Troubleshooting Data Preview</td>
<td>This can occur when an error exists in a sample set within the data source, including a different sample set than the one selected. Whenever you create or define a sample set, always click Validate to check for syntax or other errors.</td>
</tr>
<tr>
<td>In [The Sample Set Definition Screen], an error occurs when you try to preview the data.</td>
<td></td>
</tr>
<tr>
<td>Troubleshooting Different Database Types</td>
<td></td>
</tr>
<tr>
<td>Problem</td>
<td>Suggested Solution</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>The staging data source and data sources for the source data set are not all the same database type. You run aggregation (Group by) transformation discovery, but the resulting expression does not contain a MAX or MIN function as expected.</td>
<td>Check the sorting order for the staging data source and all data sources for the source data set. The sorting order must be the same. For example, the sorting order in DB2 is set by the COLLATION_SEQUENCE parameter.</td>
</tr>
<tr>
<td>Or, you run Conditional Transformation Discovery or Where Clause Discovery, but discovery yields unexpected filter matches (&gt; , &lt; , &gt;= , &lt;= ).</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 8. Glossary

This appendix provides definitions of various terms Discovery uses in connection with its statistical analysis and calculations. Whenever we have referred to these terms elsewhere in this manual, we have tried to provide references to this glossary.

**Bindings**

Binding conditions or Bindings, are like a join across, between a target table and one or more source tables, based on a natural key. The binding condition aligns the rows between the source and the target.

Binding conditions are discovered by analyzing the results of the Target Matches step (which precedes Binding Condition discovery). Target matches consist of a single column only and represent a potential Binding Condition. For example, there might be a simple equality binding condition between a single column that occurs in both source and target table. For example, both tables might have a SOCIAL_SECURITY_NUM column that can be used to align or "join" the source and target tables. In such cases, one of the target matches would be identical to the binding condition. However, in most cases a binding condition comprises several target matches, in which case the binding condition would be a composite key, for example the combination of columns like FIRST_NAME, LAST_NAME and SEX.

**Binding Selectivity**

Binding selectivity refers to the uniqueness of the values in a given binding condition. Mathematically, it is the binding cardinality divided by the number of rows in the binding condition. Binding selectivity is calculated separately on source and target sides. For example: given a binding condition

\[ \text{Source1.C1 = Target.Cx and Source2.C2 = Target.Cy} \]

the source binding selectivity is the selectivity of C1 and C2 together, in the context of the source query (the cardinality, number of unique values of C1, C2 divided by the total number of rows); and the target binding selectivity is the selectivity of Cx and Cy.

**Binding Weight**

The binding weight is used by Discovery to determine the best binding condition. The binding condition with the highest binding weight will be chosen by Discovery to be used in future steps in the discovery process. In the case where two binding conditions have the same binding weight, Discovery will choose the binding condition with the shortest length.

**Cardinality**

Cardinality refers to the number of unique values a column contains. For example, a column LAST_NAME with two Smiths, two Lees and six other unique names would have a cardinality of 8. Cardinality is closely related to Selectivity.

Not to be confused with [Relationship Cardinality](#).

**Column Match**
Column matches are potential primary foreign keys relationships. When Discovery finds statistical correlation between two columns in different tables within the same data source, it may determine the column-pair is a column match. When the statistics of the column match are higher than other column-pairs and also pass a certain threshold, Discovery will designate the column match is a primary-foreign key relationship.

Classification
In order to maximize the accuracy and reliability of data object discovery, Discovery makes use of classification at three levels: data sources, tables, and table relationships.

Correlations
Correlations are the degree to which values in source and target columns correlate to one another, given the presence of a binding condition. Remember this does not necessarily mean the values overlap or are the same, but that given a certain value in the source, there will consistently be a certain value in the target. For example, given the binding condition source SUPPLIER_ID = target SUPPLIER_ID, it may happen that whenever source REGION=ATLANTIC, then target REGION=EASTCOAST. Discovering correlations is very helpful in generating the transformation expressions you need for mapping.

Critical Data Element
A Critical Data Element, or CDE, is an optional designation for a column that identifies it as important in the project.

Marking a column as a CDE does not affect any processing or results; it simply allows you to quickly identify the column in subsequent screens. Several screens allow you to filter the view to display only CDE or non-CDE columns.

Data Object
This is a Discovery term for a group of related tables that all contain attributes of a single logical business entity, such as employees or customers. Discovery discovers data objects on the basis of primary-foreign keys among these linked tables, and uses them to assist in the efficient and accurate mapping of source and target tables.

Filters
Discovery treats filters as potential Where Clause expressions.

Joins
A join is a relationship between source tables, usually based on primary-foreign key relationships.

Non-Null Row Count
A property of tables: the total number of rows that contain data in a table, excluding Null rows.

Non-Null Selectivity
Selectivity value for a column, calculated using only the non-null rows (cardinality divided by the non-null row count).
A property of one column when compared to another; refers to the percentage of values in a column that overlap (contain similar data) in both source and target tables.

Discovery uses two thresholds to determine overlap. Whether two values overlap is determined by the correlation_character_overlap threshold, which specifies the percentage of individual characters that must be shared by the source and target values. Two values will be judged to overlap not because they are completely identical, but because they are similar to within the tolerance defined by the this threshold—which by default is set rather low, at 0.2.

The second threshold, overlap, defines the minimum number of overlapping values in a column before Discovery will consider the columns as a correlation between source and target, given some binding condition.

**Relationship Cardinality**

Refers to the data model relationship between two entities: one-to-one, one-to-many, many-to-one, or many-to-many, or data rule-related. This is a special property of maps.

**Row Count**

Total number of rows in the tables being compared, including Null rows.

**Row Hit Count**

The total number of rows with matching values when two columns are compared to each other.

Also, the total number of rows with matching values when a given binding condition is applied to a source column and a target column.

**Row Hit Rate**

The row hit count divided by the total number of rows in the column.

**Selectivity**

The degree of uniqueness of the values in a column. Mathematically, it is the Cardinality divided by the total row count; in practice, it is a useful indicator of the likelihood that a column is a potential primary key.

For example, consider a table with ten rows: a column EMPL_ID containing ten unique values would have a selectivity of 1 (10/10); a column GENDER with two possible values, M or F, would have a selectivity of .2 (2/10), and a column STATUS with all ten values the same would have a .1 selectivity (1/10).

**Sparse Column**

Any column where a large percentage of the values are the same or null. In most cases this value will be NULL or something representing null, such as '0', 'N/A', 'None,' etc.; but it could also be another value. For example, a table of employee addresses might have a STATE column, for which 90% of the values are 'NY'.

Discovery treats sparse columns differently from other columns using different discovery algorithms, as the large “near constant” incidence of one value produces incorrect transformations using standard approaches.
Discovery analyzes and detects sparse columns on the target side only. By
default, it considers as sparse any column where the most commonly
occurring value is 80% or higher, as long as the remaining (non-sparse)
values are not constant.

For all transformations, including those with low hit rates, Discovery can
derive filters that can identify and exclude the sparse values. You can
change how Discovery derives filters for sparse columns by modifying the
Task Options for the Transformations task (specifically, the tasks in the
Discover Conditional Transformations for Sparse Columns group).

Subset

Refers to a relationship between the source and target, where the source
data constitutes a subset of the target. Depending on the degree of the
match and the settings for the relevant thresholds (the subset and
subset_weight_coefficient thresholds, primarily), Discovery might interpret
a subset relationship as a valid transformation (correlation, aggregations, or
reverse pivots).

Superset

Refers to a relationship between the source and target, where the source
data constitutes a superset of the target. Depending on the degree of the
match and the settings for the relevant thresholds (the superset threshold,
primarily), Discovery might interpret a superset relationship as a valid
transformation (correlation, aggregations, or reverse pivots).

Target Match

Target matches occur across data sets, and are the starting points for
binding conditions. A target match occurs when data in a column of one
data set is related somehow to data in a column of the other data set.
(Compare to Column Match)

Target matches may be indicators of reliable binding conditions between
tables in different data sets, or they may be meaningless, accidental
overlaps.

For example, a target match between the columns LAST_NAME and
L_NAME, both of which contain text strings with high degrees of
selectivity and cardinality, will likely lead to a binding condition once it is
tested; but an equally strong match between numeric values in a column
called EMPL_ID and one called PRODUCT_ID will likely be merely
coincidental. This is why target matches are only starting points for
discovering binding conditions, and must be reinforced by correlations and
hit rates of other columns in the tables.

Transformations

Transformations are mapping expression for target columns. A
transformation expression is a SQL like statement that when applied to a
source column or columns generates the a target column.

Value Hit Count

Total number of unique data values that match when a given binding
condition is applied to a source column and a target column. This can also
be calculated as the Row Hit Count minus any rows containing a repeated
value.

The Value Hit Count is counted regardless of whether the columns or rows
also match.
This statistic is similar to the Row Hit Count except that it involves only unique values, and excludes repeats.

Value Hit Rate

The percentage of unique instances when a value in one column is present in the other. That is, it represents the Value Hit Count minus any instances when a value is repeated, divided by the Cardinality (number of unique values in the column).

This statistic is similar to the Row Hit Rate except that it involves only unique values, and excludes repeats.

Weight

Weight is a statistic calculated for binding column matches, target matches, and binding conditions. There is no set range of values and no maximum value, but as a practical matter, the higher the weight, the more reliable the match or binding condition.

Where Clause

A where clause is a filter for the source tables.
Contacting IBM

You can contact IBM for customer support, software services, product information, and general information. You also can provide feedback to IBM about products and documentation.

The following table lists resources for customer support, software services, training, and product and solutions information.

Table 36. IBM resources

<table>
<thead>
<tr>
<th>Resource</th>
<th>Description and location</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Support Portal</td>
<td>You can customize support information by choosing the products and the topics that interest you at <a href="http://www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server">www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server</a></td>
</tr>
<tr>
<td>Software services</td>
<td>You can find information about software, IT, and business consulting services, on the solutions site at <a href="http://www.ibm.com/businesssolutions/">www.ibm.com/businesssolutions/</a></td>
</tr>
<tr>
<td>My IBM</td>
<td>You can manage links to IBM Web sites and information that meet your specific technical support needs by creating an account on the My IBM site at <a href="http://www.ibm.com/account/">www.ibm.com/account/</a></td>
</tr>
<tr>
<td>Training and certification</td>
<td>You can learn about technical training and education services designed for individuals, companies, and public organizations to acquire, maintain, and optimize their IT skills at <a href="http://www.ibm.com/software/sw-training/">http://www.ibm.com/software/sw-training/</a></td>
</tr>
</tbody>
</table>

Providing feedback

The following table describes how to provide feedback to IBM about products and product documentation.

Table 37. Providing feedback to IBM

<table>
<thead>
<tr>
<th>Type of feedback</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product feedback</td>
<td>You can provide general product feedback through the Consumability Survey at <a href="http://www.ibm.com/software/data/info/consumability-survey">www.ibm.com/software/data/info/consumability-survey</a></td>
</tr>
</tbody>
</table>
Table 37. Providing feedback to IBM (continued)

<table>
<thead>
<tr>
<th>Type of feedback</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documentation feedback</td>
<td>To comment on the information center, click the Feedback link on the top right side of any topic in the information center. You can also send comments about PDF file books, the information center, or any other documentation in the following ways:</td>
</tr>
<tr>
<td></td>
<td>• Online reader comment form: <a href="http://www.ibm.com/software/data/rcf/">www.ibm.com/software/data/rcf/</a></td>
</tr>
<tr>
<td></td>
<td>• E-mail: <a href="mailto:comments@us.ibm.com">comments@us.ibm.com</a></td>
</tr>
</tbody>
</table>
Product accessibility

You can get information about the accessibility status of IBM products.

The IBM InfoSphere Information Server product modules and user interfaces are not fully accessible. The installation program installs the following product modules and components:

• IBM InfoSphere Business Glossary
• IBM InfoSphere Business Glossary Anywhere
• IBM InfoSphere DataStage
• IBM InfoSphere FastTrack
• IBM InfoSphere Information Analyzer
• IBM InfoSphere Information Services Director
• IBM InfoSphere Metadata Workbench
• IBM InfoSphere QualityStage™

For information about the accessibility status of IBM products, see the IBM product accessibility information at [http://www.ibm.com/able/product_accessibility/index.html](http://www.ibm.com/able/product_accessibility/index.html).

Accessible documentation

Accessible documentation for InfoSphere Information Server products is provided in an information center. The information center presents the documentation in XHTML 1.0 format, which is viewable in most Web browsers. XHTML allows you to set display preferences in your browser. It also allows you to use screen readers and other assistive technologies to access the documentation.

IBM and accessibility

See the [IBM Human Ability and Accessibility Center](http://www.ibm.com/able/) for more information about the commitment that IBM has to accessibility:
Accessing product documentation

Documentation is provided in a variety of locations and formats, including in help that is opened directly from the product client interfaces, in PDF files, and in HTML files.

Obtaining the documentation

The documentation is distributed with the product, can be accessed from a Web browser, and is orderable.

- PDF file books are available online and periodically refreshed at [www.ibm.com/support/docview.wss?uid=swg27020315](http://www.ibm.com/support/docview.wss?uid=swg27020315)
- You can also order IBM publications in hardcopy format online or through your local IBM representative. To order publications online, go to the IBM Publications Center at [http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss](http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss)

Providing feedback about the documentation

You can send your comments about documentation in the following ways:

- E-mail: comments@us.ibm.com
Links to non-IBM Web sites

This information center may provide links or references to non-IBM Web sites and resources.

IBM makes no representations, warranties, or other commitments whatsoever about any non-IBM Web sites or third-party resources (including any Lenovo Web site) that may be referenced, accessible from, or linked to any IBM site. A link to a non-IBM Web site does not mean that IBM endorses the content or use of such Web site or its owner. In addition, IBM is not a party to or responsible for any transactions you may enter into with third parties, even if you learn of such parties (or use a link to such parties) from an IBM site. Accordingly, you acknowledge and agree that IBM is not responsible for the availability of such external sites or resources, and is not responsible or liable for any content, services, products or other materials on or available from those sites or resources.

When you access a non-IBM Web site, even one that may contain the IBM-logo, please understand that it is independent from IBM, and that IBM does not control the content on that Web site. It is up to you to take precautions to protect yourself from viruses, worms, trojan horses, and other potentially destructive programs, and to protect your information as you deem appropriate.
Notices and trademarks

This information was developed for products and services offered in the U.S.A.

Notices

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites.
sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to
IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Trademarks**

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

The following terms are trademarks or registered trademarks of other companies:

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office

UNIX is a registered trademark of The Open Group in the United States and other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The United States Postal Service owns the following trademarks: CASS, CASS Certified, DPV, LACSLink, ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS and United States Postal Service. IBM Corporation is a non-exclusive DPV and LACSLink licensee of the United States Postal Service.

Other company, product or service names may be trademarks or service marks of others.
### Index

#### A
- **active window** (for Discovery Engine Service) 10
- **Activity Viewer** 85
- **Activity Pane** 80
canceling tasks 94
definition 100
descending bar 92
aggregations, discovering 523
ambiguous binding condition 526
approving Bindings 524
maps 69
PF Keys 317
tables 318
where clauses 538
approving a map 502
architecture
Discovery installation examples 6
arrow mode (viewing PF Keys) 323

#### B
- binding condition ambiguous 526
weight 525
binding selectivity, definition 749
binding weight, definition 749
bindings
re-running 521
verifying 524
Bindings
defining manually 522
definition 749
editing 523
hits and misses 531
overview 36
reports on 618
selecting primary 525
viewing and managing 526
bindings step 524
Blank Count, definition 635
blue icons in diagrams 137
Bound Rows Passing Filter grid column, definition 635
boundaries, dragging to resize 130
business solutions provided by InfoSphere Discovery 3

#### C
cancelling a task 94
cannot connect 744
cardinality, definition 749
case sensitivity
in Discovery Studio 15
case statements 41
Character Correlation Overlap threshold, definition 749
table relationship tables in Data Objects 344
cancelling tasks 94
definition 100
progress bar 92
aggregations, discovering 523
ambiguous binding condition 526
approving Bindings 524
maps 69
PF Keys 317
tables 318
where clauses 538
approving a map 502
architecture
Discovery installation examples 6
arrow mode (viewing PF Keys) 323

#### D
data dirty 713
Data Inventory tab 365
Data Objects
approving 360
create related sample sets 341
data set tabs 359
define manually 338
deleting 352
deleting child tables 355
discovering 60, 342
Generate and Regenerate options 343
include attribute and reference tables within 344
join statistics 363
manually adding child tables 353
manually adding to data set 348
overview 60
review 358
root table 360
Skip Columns 323
step 358
tools 363
view as diagram 361
Data Preview
common UI elements 112
definition 111
data rule data sets
modifying 216
supported databases 172
Data Rule Key, definition 565
data rules
changing names 572
data types in shared lookup tables 573
exporting lookup table 568
generate 562
grid column 635
list 570
recreating after importing a project 170
shared lookup tables 573
data set
changing scan formats 160
copy rows
with hidden bands or columns 142
Correctly Transformed Rows (Hits)
Passing Filter, definition 635
correlations
definition 749
Critical Data Element (CDE) 283, 749
cross-reference tables in Data Object generation 345
Custom Option Sets 711
customer support 755
CWM (Common Warehouse Metamodel) 605

### Copy Rows

<table>
<thead>
<tr>
<th>Classification</th>
<th>Changing the Data Set</th>
<th>Data Sets</th>
<th>PF Keys</th>
<th>Tables</th>
<th>Classification Propagation</th>
<th>Progress Bar</th>
<th>Canceling Tasks</th>
<th>Definition</th>
<th>Approving Bindings</th>
<th>Maps</th>
<th>PF Keys</th>
<th>Tables</th>
<th>Where Clauses</th>
<th>Approving a Map</th>
<th>Architecture</th>
<th>Discovery Installation Examples</th>
<th>Arrow Mode (Viewing PF Keys)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>B</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>C</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 2006, 2011
Data Sets
- identify tables and columns used in target column 387
- original tables and files 172
- related sample sets with aliases 341
- tabs in steps 315

Data Sets summary 57

Data sources
- data rule data sets 13
- importing text files 194
- staging data source 159

Data Type, definition 635

Data types
- DB2 722
- MSSQL Server 724
- Oracle 10g and 9i 723
- Sybase 726
- Teradata 725
- data, masking confidential data 253
- database report 279
- database vendors, supported 11
- datetimestring data type, changing format 268
- DB2 importing long numbers as VARCHAR 119
- Debug Log 730
- default expressions in data rules 558
- default names
  - Data Objects 340
  - duplicate Data Objects 340
  - Target Maps 489
- deleting
  - Discovery projects 167
  - logical tables from a project 222, 223
  - deleting physical and logical tables 222
  - delimited text files, importing 196
  - dependent sample set 261
  - depth of Data Objects 345, 346
  - diagram panes
    - Data Objects 361
    - overview for all screens 136
  - diagrams
    - printing and saving 140
  - Discovery Engine configuration file 730
  - in Task Manager 727
- Discovery Engine Service
  - ODBC connection 12
  - spawning Discovery Engines 9
- Discovery Quick Reference 501
- Discovery Server
  - connecting to a different server 162
  - definition 7
    - JDBC connection to repository 12
- Discovery Server Configuration Utility 734
- Discovery Server in Task Manager 733
- Discovery Server repository changing 734
- Discovery Studio
  - sequence of work 51
  - timeout (lost connection to Discovery Server) 165
- Discovery XML export 602

E
- editing expressions 115
- Error List
  - definition 131
  - Error Log 730
- error logs
  - Discovery Engine log viewer 103
  - Discovery Server log viewer 105
- Excel version required for reports 607, 608
- Exception Manager
  - creating a new job and executable 619
  - executing jobs 632
  - validation jobs 620
- Exception Manager repository changing 734
- Expression Editor 115
- Expression, definition 635
- expressions
  - format of PF Key expression 295
  - format of Unified Schema PF Key expression 456
  - invalid due to deleting tables 222

F
- filters
  - clear 146
  - define using Where tab 536
  - in list panes, overview for all screens 135
  - preview data 108
  - tasks in Activity Viewer 88
  - fixed-width text files, importing 202
  - Foreign Keys (PF Keys) 332
  - format of discovered dates and numbers 17
  - Full Expand in Group By 153
  - fuzzy matches 550
  - fuzzy matching for strings 17

G
- generate Reverse Pivot maps 579
- grids
  - grouping by columns 151
  - icons, overview for all screens 151
  - overview for all screens 140
  - primary expressions 148
  - group numbers 432

H
- heartbeat 734
- hit count
  - see row hit count 749
- hits and misses
  - exporting 595
  - for Data Objects 363
  - for PF keys 334

I
- IBM InfoSphere Exception Manager
  - primary key column 225
  - importing
    - full project 170
    - full projects 171
    - primary-foreign keys 176
    - project objects 171
    - recreating staging data source 170
  - Incorrectly Transformed Rows (Hits) passing Filter, definition 635
  - inner join, creating 511

J
- JDBC connection 163, 186
- JDBC URL 163, 186
- jobs
  - Exception Manager 620
  - join condition, see Joins 514
  - Joins
    - defining in Join Condition grid 510
    - defining in Join Matches Window 511
    - defining manually 510
    - defining with drag and drop 511
    - discovering 508
    - editing 512
    - grid tools 514
    - Joins tab 512
    - overview 506
    - re-run discovery 509
    - reviewing 512
    - statistics 514
    - view hits and misses 516
    - viewing 516
  - Joins Columns grid 514

L
- legal notices 763
- Length Frequencies in Column Analysis 276
- list panes, overview for all screens 133
- locked objects 84
- locked projects 165
- log
  - debug 730
  - error 730
- log files
  - Discovery Engine Services 11
  - location and content 101
  - viewing 101
- log viewer
  - Discovery Engine Log Viewer 101
  - Discovery Server Log Viewer 105
logical column 233
logical table
  definition 226
deleted along with physical table 223
deleting physical table 222
log tables 175
deleting 222, 223
join condition (From) 233
logs
  trace logs 101
lookup tables
  importing 216

M
main menus, overview for all screens 117
Manage Column Matches 311
Map Cardinality, definition 635
Map Selectivity, definition 635
maps
  approval checkboxes 69
  approving a map 502
  defining manually 496
  deleting a source table 499
discovering 72, 494
  Information pane 503
  manually adding a source table 499
Maps sub-tabs, overview 66
  no maps discovered 495
  overview 72, 493
Query pane 504
reviewing 501
searching in preview data 651
Summary diagram 505
Summary tab 502
Transformations tab 546
viewing map SQL 504
Maps list, overview for all Maps tabs 68
Match and Merge
  group numbers 432
  matches
    fuzzy matches 550
MDM, custom option set 715
messages, viewing in Discovery Studio 100
migrating data
  loading target data 606
  misses
    export Bindings misses 527
    export transformation misses 555
Misses, definition 635
Mode, definition 635
Mode%, definition 635

N
Name column definition 635
names
  valid names for projects 163
  native type, definition on Column Analysis tab 635
natural joins
  see Joins 507
non-IBM Web sites
  links to 761

O
ODBC
collection to staging data source 12
  non-supported connections 163
  ODBC buffer, maximum field size 14
  ODBC driver, MSSQL Server 268
Only Refresh Statistics 70
optimizing composite PF key discovery 290
options
  changing option values 82
  default option set 711
  edit option set 82
  profile options 659
  read-only option sets 82
  task options 667
orange icons in diagrams 137
Output Pane
  definition 132
  icons 131
  overlap
    fuzzy matches 550
Over, definition 749

P
panes
  autohide pane 125
parallel profiling 15
passwords
  adding project password 167
  changing project password 167
  removing project password 167
setting project password 163
permission for text files 175
PF key discovery on large tables 290
PF keys
  composite PF keys 285
PF Keys
  defining (adding) manually 293
  defining manually 300
  deleting all PF Keys and column matches between tables 303
  deleting all PF Keys between tables 303
diagram pane 319
diagram pane tools 323
discovering 288
display modes in the diagram 326
editing 302
form of inclusion dependency 285
how Discovery creates PF Keys 59
overview 314
re-run discovery 291
PF Keys (continued)
  reading the Connected Tables list 315
  relationship classification,
    changing 295
    defined 178
    reviewing 313
    Show Duplicates 335
    Show PF Keys (PF Keys) 329
  skipping discovery 288
physical columns 234
physical table
  definition 227
deleting 222
PID (Windows process ID) 104
pivot columns, also see reverse pivot discovering 577
Precision
  in numberstring format 265
  on Column Analysis tab 635
primary
  changing the primary 148
  primary key column 225
  primary key icon (PF Keys step) 651
  primary sample set
    what to do after changing 216
primary-foreign keys
  importing 176
printing diagrams 140
Processing Options dialog, definition 76
product accessibility
  accessibility 757
product documentation
  accessing 759
profiling
  default method 15
  definition 15
  when to reprofile tables 15
profiling, definition 260
progress bar, Activity Viewer 92
projects
  about 158
  changing project type 159
  changing properties 167
  changing scan formats 160, 168
  changing the staging data source 169
  cleaning up the staging data source 170
  closing 167
  creating 56, 158, 163
  definition 56
  deleting 167
  importing a full project 171
  importing project objects 171
open projects 161
opening 163, 165
read-only 165, 166
saving 166
TD and SDD defined 25
types 25, 159
unlocking 166
valid names 163
variables 170
where stored 159
without maps 173
propagating approved classifications 18
properties
of Data Objects 351
of projects, changing 167

Q
quitting Discovery Studio 51

R
Re-Run Step
button 125
read permission 175
rarrange table column order 387
rarrange target column order 387
reference tables in Data Objects 344
refresh statistics
stale statistics 651
Refresh Statistics, definition 70
related sample sets
default names 341
Relationship Cardinality, definition 749
relationships, types discovered 726
reload column matches 330
Remove Default (Data Rules),
definition 558
reporting from the Column Analysis
grid 279

S
sample sets 180
changing the primary 175
creating 238
defining 238
deleting 238
dependent sample sets 261
Full Set 175, 180
preview data 244
primary sample set 235, 237
restaging 261
saving
Discovery projects 166
saving diagrams 140
Scale, definition 635
scan formats
about 160
changing 168
SDD project, defined 25
SDD projects 25
search
in grids 651
see Source Data Discovery projects 25
Selectivity
definition 749
Selectivity, definition 635
sensitive data repository (SDR) 18
serial profiling 15
Server Port 50
services
in Task Manager 727, 733
Show Hits (PF Keys, Column Matches) 337
skipping steps
when generating maps 69
software services 755
sort order
revert to original column sort order 150
Source Bound Selectivity, definition 635
Source Column, definition 635
Source Data Discovery projects 25, 365
Source Hits, definition 635
sparse column, definition 749
Sparse, definition 635
stage sample sets 261
staging
dependent sample sets 261
staging data source
changing the default 169
cleaning up 170
default 119
Drop Existing Table 596
dropping tables 159
managing 159
staging tables 52
staging, explicit, definition 260
starting
Discovery Engine processes 728
Discovery Engine Service 727
Discovery Server 733
Discovery Studio 737
starting Discovery Studio 50
status icons, overview for all screens 650
steps, re-running 54
stopping
Discovery Engines 728
Discovery Server 733
Discovery Studio 737
stopping Discovery Studio 51
subset
fuzzy matches 550
Subset, definition 749
superset
fuzzy matches 550
Superset, definition 749
support
customer 755
tables
adding to data sets 186
changing names during import 192
dropping in the staging data source 170
PF Key classification 320
Target Bound Selectivity, definition 635
target column
identify columns in target column 387
Target Matches
data preview 492
defining manually 477
definition 749
diagram 489
generating 473
list 488
overview 473
statistics 491
target table
copy 394
rearrange column order 387
tasks
cancel a task 94
deleting from the Activity Viewer 97
running tasks in multiple projects 165
view summary of task activity 92
TD project, defined 25
TD projects
see Transformation Discovery projects 25
text files
add to data set 218
combining into one table 208
cleaned to tables after import 172
directory access 175
Discovery cannot access 175
point to a different text file 219
Skip Columns 205
thresholds, see options 82
Total Rows Passing Filter, definition 635
trademarks
list of 763
transformations
approving 550
complex 41
defining manually 546
grid tools 550
primary 549
transformations between dates and numbers 542
troubleshooting, general 741
truncation
errors 542
of data when importing text files 194, 207
of empty spaces when importing text files 199, 205
of numbers with long precision in DB2 119
when masking values 240
when value is longer than column length 253
types of projects 25
U
Un-approve Step
definition 124
Unbound Rows Passing Filter,
definition 635
Unconnected Tables, definition 317
Unified Column Analysis
  grid tools 424
  statistics 429
Unified Schema PF Keys
diagram pane 466
  display modes in the diagram 467
  editing 460
  overview 463
  reading the Connected Tables list 464
  reviewing 463
  Show Duplicates 470
User Name in Notes 119
  using Discovery Studio 51

V
Validate Step, definition 124
validation jobs
  creating 620
  executing 632
  location of executable 630
  primary key column 225
  updating 631
  using variables 170
Value Frequencies in Column Analysis 273, 274
Value Hit count, definition 749
Value Hit Rate, definition 635, 749
VARCHAR
  importing long numbers as VARCHAR 119
  variables
    for validation jobs 170
    in sample sets 243

W
Web sites
  non-IBM 761
weight of binding condition 525
Weight, definition 635, 749
where clauses
  how created 531
  primary 537
window panes, overview for all screens 125