Informatica® Data Quality
10.5

Data Quality Getting Started Guide

Informatica Data Quality Data Quality Getting Started Guide
10.5
March 2021

Table of Contents

Preface

Read the *Data Quality Getting Started Guide* to discover the main features and functionality of Data Quality and to learn how to perform data quality tasks in Informatica Developer and Informatica Analyst.

Informatica Resources

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

Informatica Network

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

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

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

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

Informatica Documentation

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

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

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

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

You can find Informatica Velocity resources at http://velocity.informatica.com. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

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

Informatica Global Customer Support

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

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

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

CHAPTER 1

Getting Started Overview

This chapter includes the following topics:

Informatica Domain Overview

Informatica has a service-oriented architecture that provides the ability to scale services and to share resources across multiple machines. The Informatica domain is the primary unit for the management and administration of services.

You can log in to Informatica Administrator after you install Informatica. You use the Administrator tool to manage the domain and configure the required application services before you can access the remaining application clients.

The Informatica domain contains the following components:

- Application clients. A group of clients that you use to access underlying Informatica functionality. Application clients make requests to the Service Manager or application services.
- Application services. A group of services that represent server-based functionality. An Informatica domain can contain a subset of application services. You create and configure the application services that the application clients require.

  Application services include system services that can have a single instance in the domain. When you create the domain, the system services are created for you. You can configure and enable a system service to use the functionality that the service provides.
- Profile warehouse. A relational database that the Data Integration Service uses to store profile results.
- Reference data warehouse. A relational database that stores reference data values for the reference table objects in the Model repository.
- Repositories. A group of relational databases that store metadata about objects and processes required to handle user requests from application clients.
- Service Manager. A service that is built in to the domain to manage all domain operations. The Service Manager runs the application services and performs domain functions including authentication, authorization, and logging.

- Workflow database. A relational database that stores run-time metadata for workflows.

The following table lists the application clients, not including the Administrator tool, and the application services and the repositories that the client requires:

Application Client	Application Services	Repositories
Informatica Analyst	- Analyst Service - Content Management Service - Data Integration Service - Model Repository Service - Search Service	Model repository
Informatica Developer	- Analyst Service - Content Management Service - Data Integration Service - Model Repository Service	Model repository
Metadata Manager	- Metadata Manager Service - PowerCenter Integration Service - PowerCenter Repository Service	- Metadata Manager repository - PowerCenter repository
PowerCenter® Client	- PowerCenter Integration Service - PowerCenter Repository Service	PowerCenter repository
Web Services Hub Console	- PowerCenter Integration Service - PowerCenter Repository Service - Web Services Hub	PowerCenter repository

The following application services are not accessed by an Informatica application client:

- PowerExchange® Listener Service. Manages the PowerExchange Listener for bulk data movement and change data capture. The PowerCenter Integration Service connects to the PowerExchange Listener through the Listener Service.

- PowerExchange Logger Service. Manages the PowerExchange Logger for Linux, UNIX, and Windows to capture change data and write it to the PowerExchange Logger Log files. Change data can originate from DB2 recovery logs, Oracle redo logs, a Microsoft SQL Server distribution database, or data sources on an i5/OS or z/OS system.

- SAP BW Service. Listens for RFC requests from SAP BI and requests that the PowerCenter Integration Service run workflows to extract from or load to SAP BI.

Feature Availability

Informatica products use a common set of applications. The product features that you can use depend on your product license.

The following table describes the licensing options and the application features available with each option:

Licensing Option	Informatica Developer Features	Informatica Analyst Features
Data Quality	- Create and run mappings - Create and run mapplets and rules - Create and run profiles, including profiles for enterprise discovery, primary key and foreign key discovery, and functional dependency discovery - Curate inferred profile results - Create and run scorecards - Manage reference tables - Identify the exception records in a data source - Export objects to PowerCenter	- Profiling, including enterprise discovery Use discovery search to find where data and metadata exist in the profiling repositories - Create and run scorecards - Curate inferred profile results - Create and run profiling rules - Manage reference tables - Create rule specifications and compile rule specifications into mapplets - Review and edit exception records
Data Services	- Create logical data object models - Create and run mappings with Data Services transformations - Create SQL data services - Create web services - Export objects to PowerCenter	- Manage reference tables
Data Services and Profiling Option	- Create logical data object models - Create and run mappings with Data Services transformations - Create SQL data services - Create web services - Export objects to PowerCenter - Create and run rules with Data Services transformations - Perform profiling	- Manage reference tables

Note: If you use Informatica products with a Data Engineering licence, you might experience a different set of features. For example, Data Engineering applications do not integrate with PowerCenter. You cannot perform exception record management in Data Engineering Quality.

Introducing Informatica Analyst

Informatica Analyst is a web-based application client that analysts can use to analyze, cleanse, standardize, profile, and score data in an enterprise.

Depending on your license, business analysts and developers use the Analyst tool for data-driven collaboration. You can perform column and rule profiling, scorecarding, and bad record and duplicate record management. You can also manage reference data and provide the data to developers in a data quality solution.

Informatica Developer Overview

The Developer tool is an application that you use to design and implement data integration, data quality, data profiling, data services, and data engineering solutions.

You can use the Developer tool to import metadata, create connections, and create data objects. You can also use the Developer tool to create and run profiles, mappings, and workflows.

Informatica Developer Views

The Developer tool workbench includes an editor and views. You edit objects, such as mappings, in the editor. The Developer tool displays views based on which object is selected in the editor.

You can select additional views, hide views, and move views to another location in the Developer tool workbench.

To select the views you want to display, click **Window** > **Show View**.

The Developer tool displays the following views by default:

Connection Explorer view

Displays connections to relational databases.

Data Viewer view

Displays source data, profile results, and previews the output of a transformation.

Object Explorer view

Displays the domain and the design-time and run-time objects in the domain. Design-time objects are stored in projects and folders in the Model repository. Run-time objects are stored as part of a run-time application on a Data Integration Service.

Outline view

Displays objects that are dependent on an object selected in the **Object Explorer** view.

Progress view

Displays the progress of operations in the Developer tool, such as a mapping run.

Properties view

Displays the properties for an object that is selected in the editor.

You can also use the **Show View** menu to show the following views:

Alerts view

Displays connection status alerts.

Checked Out Objects view

Displays all objects that you have checked out.

Notifications view

Displays options to notify users or groups when all work in the Human task is complete.

Object Dependencies view

Displays object dependencies when you view, modify, or delete an object.

Search view

Displays the search results. You can also launch the search options dialog box.

Tags view

Displays tags that define an object in the Model repository based on business usage.

Informatica Developer Welcome Page

The Welcome page appears the first time that you open the Developer tool. Use the Welcome page to learn how to set up and start working in the Developer tool.

The Welcome page displays the following options:

- Overview. Click the Overview button to get an overview of data quality and data services solutions.
- First Steps. Click the First Steps button to learn more about setting up the Developer tool and accessing Informatica Data Quality and Informatica Data Services lessons.
- Tutorials. Click the Tutorials button to see tutorial lessons for data quality and data services solutions.
- Web Resources. Click the Web Resources button for a link to the Informatica Knowledge Base. You can access the Informatica How-To Library. The Informatica How-To Library contains articles about Informatica Data Quality, Informatica Data Services, and other Informatica products.
- Workbench. Click the Workbench button to start working in the Developer tool.

Click **Help** > **Welcome** to access the welcome page after you close it.

Cheat Sheets

The Developer tool includes cheat sheets as part of the online help. A cheat sheet is a step-by-step guide that helps you complete one or more tasks in the Developer tool.

When you complete a cheat sheet, you complete the tasks and see the results. For example, after you complete a cheat sheet to import and preview a relational data object, you have imported a relational database table and previewed the data in the Developer tool.

To access cheat sheets, click **Help** > **Cheat Sheets**.

Data Quality and Profiling

Use the data quality capabilities in the Developer tool to analyze the content and structure of your data. You can enhance the data in ways that meet your business needs.

Use the Developer tool to design and run processes that achieve the following objectives:

- Profile data. Profiling reveals the content and structure of your data. Profiling is a key step in any data project as it can identify strengths and weaknesses in your data and help you define your project plan.

- Create scorecards to review data quality. A scorecard is a graphical representation of the quality measurements in a profile.
- Standardize data values. Standardize data to remove errors and inconsistencies that you find when you run a profile. You can standardize variations in punctuation, formatting, and spelling. For example, you can ensure that the city, state, and ZIP code values are consistent.
- Parse records. Parse data records to improve record structure and derive additional information from your data. You can split a single field of freeform data into fields that contain different information types. You can also add information to your records. For example, you can flag customer records as personal or business customers.
- Validate postal addresses. Address validation evaluates and enhances the accuracy and deliverability of your postal address data. Address validation corrects errors in addresses and completes partial addresses by comparing address records against reference data from national postal carriers. Address validation can also add postal information that speeds mail delivery and reduces mail costs.
- Find duplicate records. Duplicate record analysis compares a set of records against each other to find similar or matching values in selected data columns. You set the level of similarity that indicates a good match between field values. You can also set the relative weight fixed to each column in match calculations. For example, you can prioritize surname information over forename information.
- Create and run data quality rules. Informatica provides pre-built rules that you can run or edit to suit your project objectives. You can create rules in the Developer tool.
- Collaborate with Informatica users. The rules and reference data tables you add to the Model repository are available to users in the Developer tool and the Analyst tool. Users can collaborate on projects, and different users can take ownership of objects at different stages of a project.
- Export mappings to PowerCenter. You can export mappings to PowerCenter to reuse the metadata for physical data integration or to create web services.

The Tutorial Story

HypoStores Corporation is a national retail organization with headquarters in Boston and stores in several states. It integrates operational data from stores nationwide with the data store at headquarters on regular basis. It recently opened a store in Los Angeles.

The headquarters includes a central ICC team of administrators, developers, and architects responsible for providing a common data services layer for all composite and BI applications. The BI applications include a CRM system that contains the master customer data files used for billing and marketing.

HypoStores Corporation must perform the following tasks to integrate data from the Los Angeles operation with data at the Boston headquarters:

- Examine the Boston and Los Angeles data for data quality issues.
- Parse information from the Los Angeles data.
- Standardize address information across the Boston and Los Angeles data.
- Validate the accuracy of the postal address information in the data for CRM purposes.

The Tutorial Structure

The *Getting Started Guide* contains tutorials that include lessons and tasks.

Lessons

Each lesson introduces concepts that will help you understand the tasks to complete in the lesson. The lesson provides business requirements from the overall story. The objectives for the lesson outline the tasks that you will complete to meet business requirements. Each lesson provides an estimated time for completion. When you complete the tasks in the lesson, you can review the lesson summary.

If the environment within the tool is not configured, the first lesson in each tutorial helps you do so.

Tasks

The tasks provide step-by-step instructions. Complete all tasks in the order listed to complete the lesson.

Informatica Analyst Tutorial

During this tutorial, an analyst logs into the Analyst tool, creates projects and folders, creates profiles and rules, scores data, and creates reference tables.

The lessons you can perform depend on whether you have the Informatica Data Quality or Informatica Data Services products.

The following table describes the lessons you can perform, depending on your product:

Lesson	Description	Product
Lesson 1. Setting up Informatica Analyst	Log in to the Analyst tool and create a project and folder for the tutorial lessons.	Data Quality Data Services
Lesson 2. Creating Data Objects	Import a flat file as a data object and preview the data.	Data Quality
Lesson 3. Creating Quick Profiles	Creating a quick profile to quickly get an idea of data quality.	Data Quality
Lesson 4. Creating Custom Profiles	Create a custom profile to configure columns, and sampling and drilldown options.	Data Quality
Lesson 5. Creating Expression Rules	Create expression rules to modify and profile column values.	Data Quality
Lesson 6. Creating and Running Scorecards	Create and run a scorecard to measure data quality progress over time.	Data Quality
Lesson 7. Creating Reference Tables from Profile Results	Create a reference table that you can use to standardize source data.	Data Quality Data Services
Lesson 8. Creating Reference Tables	Create a reference table to establish relationships between source data and valid and standard values.	Data Quality Data Services

Informatica Developer Tool

In this tutorial, you use the Developer tool to perform several data quality operations.

Informatica Data Quality users use the Developer tool to design and run processes that enhance data quality. Informatica Data Quality users also use the Developer tool to create and run profiles that analyze the content and structure of data.

Complete the following lessons in the data quality tutorial:

Lesson 1. Setting Up Informatica Developer

Create a connection to a Model repository that is managed by a Model Repository Service in a domain. Create a project and folder to store work for the lessons in the tutorial. If the domain includes more than one Data Integration Service, select a service.

Lesson 2. Importing Physical Data Objects

You will define data quality processes for the customer data files associated with these objects.

Lesson 3. Profiling Data

Profiling reveals the content and structure of your data.

Profiling includes join analysis, a form of analysis that determines if a valid join is possible between two data columns.

Lesson 4. Parsing Data

Parsing enriches your data records and improves record structure. It can find useful information in your data and also derive new information from current data.

Lesson 5. Standardizing Data

Standardization removes data errors and inconsistencies found during profiling.

Lesson 6. Validating Address Data

Address validation evaluates the accuracy and deliverability of your postal addresses and fixes address errors and omissions in addresses.

Tutorial Prerequisites

Before you can begin the tutorial lessons, the Informatica domain must be running with at least one node set up.

The installer includes tutorial files that you will use to complete the lessons. You can find all the files in both the client and server installations:

- You can find the tutorial files in the following location in the Developer tool installation path:

  ```
  <Informatica Installation Directory>\clients\DeveloperClient\Tutorials
  ```

- You can find the tutorial files in the following location in the services installation path:

  ```
  <Informatica Installation Directory>\server\Tutorials
  ```

You need the following files for the tutorial lessons:

- All_Customers.csv
- Boston_Customers.csv
- LA_customers.csv

Part I: Getting Started with Informatica Analyst

This part contains the following chapters:

Lesson 1. Setting Up Informatica Analyst

This chapter includes the following topics:

Setting Up Informatica Analyst Overview

Before you start the lessons in this tutorial, you must set up the Analyst tool. To set up the Analyst tool, log in to the Analyst tool and create a project and a folder to store your work.

The Informatica domain is a collection of nodes and services that define the Informatica environment. Services in the domain include the Analyst Service and the Model Repository Service. The Analyst Service runs the Analyst tool, and the Model Repository Service manages the Model repository. When you work in the Analyst tool, the Analyst tool stores the assets that you create in the Model repository.

You must create a project before you can create assets in the Analyst tool. A project contains assets in the Analyst tool. A project can also contain folders that store related assets, such as data objects that are part of the same business requirement.

Objectives

In this lesson, you complete the following tasks:

- Log in to the Analyst tool.
- Create a project to store the assets that you create in the Analyst tool.
- Create a folder in the project that can store related assets.

Prerequisites

Before you start this lesson, verify the following prerequisites:

- An administrator has configured a Model Repository Service and an Analyst Service in the Administrator tool.

- You have the host name and port number for the Analyst tool.
- You have a user name and password to access the Analyst Service. You can get this information from an administrator.

Timing

Set aside 5 to 10 minutes to complete this lesson.

Task 1. Log In to Informatica Analyst

Log in to the Analyst tool to begin the tutorial.

1. Start a Microsoft Internet Explorer or Google Chrome browser.
2. In the Address field, enter the URL for Informatica Analyst:

```
http[s]://<fully qualified host name>:<port number>/analyst
```

3. If the domain uses LDAP or native authentication, enter your user name and password on the login page.
4. Select **Native** or the name of a specific security domain.

   The Security Domain field appears when the Informatica domain uses LDAP or Kerberos authentication. If you do not know the security domain that your user account belongs to, contact the Informatica domain administrator.
5. Click **Log In**.

   The Analyst tool opens on the **Start** workspace.

Task 2. Create a Project

In this task, you create a project to contain the assets that you create in the Analyst tool. Create a tutorial project to contain the folder for the project.

1. On the **Manage** header, click **Projects**.

   The **Projects** workspace appears.
2. From the **Actions** menu, click **New** > **Project**.

   The **New Project** window appears.
3. Enter your name prefixed by "Tutorial_" as the name of the project.
4. Click **OK**.

Task 3. Create a Folder

In this task, you create a folder to store related assets. You can create a folder in a project or another folder. Create a folder named Customers to store the assets related to the data quality project.

1. In the **Projects** panel, select the tutorial project.

2.   From the **Actions** menu, click **New** > **Folder**.

The **New Folder** window appears.

3.   Enter Customers for the folder name.

4.   Click **OK**.

The folder appears under the tutorial project.

Setting Up Informatica Analyst Summary

In this lesson, you learned that the Analyst tool stores assets in projects and folders. A Model repository contains the projects and folders. The Analyst Service runs the Analyst tool. The Model Repository Service manages the Model repository. The Analyst Service and the Model Repository Service are application services in the Informatica domain.

You logged in to the Analyst tool and created a project and a folder.

Now, you can use the Analyst tool to complete other lessons in this tutorial.

CHAPTER 3

Lesson 2. Creating Data Objects

This chapter includes the following topics:

Creating Data Objects Overview

In the Analyst tool, a data object is a representation of data based on a flat file or relational database table. You create a flat file or table object and then run a profile against the data in the flat file or relational database table. When you create a flat file data object in the Analyst tool, you can upload the file to the flat file cache on the machine that runs the Analyst tool or you can specify the network location where the flat file is stored.

Story

HypoStores keeps the Los Angeles customer data in flat files. HypoStores needs to profile and analyze the data and perform data quality tasks.

Objectives

In this lesson, you complete the following tasks:

1. Upload the flat file to the flat file cache location and create a data object.
2. Preview the data for the flat file data object.

Prerequisites

Before you start this lesson, verify the following prerequisites:

- You have completed lesson 1 in this tutorial.
- You have the LA_Customers.csv flat file. You can find this file in the <Installation Root Directory>\<Release Version>\clients\DeveloperClient\Tutorials folder.

Timing

Set aside 5 to 10 minutes to complete this task.

Task 1. Create the Flat File Data Objects

In this task, you create a flat file data object from the LA_Customers file.

1. In the Analyst tool, click **New** > **Flat File Data Object**.

    The **Add Flat File** wizard appears.

2. Select **Browse and Upload**, and click **Browse**.

3. Browse to the location of LA_Customers.csv, and click **Open**.

4. Click **Next**.

    The **Choose type of import** panel displays **Delimited** and **Fixed-width** options. Select the **Delimited** option. The default option is **Delimited**.

5. Click **Next**.

6. Under **Specify the delimiters and text qualifiers used in your data**, select **Double quotes** as a text qualifier.

7. Under **Specify lines to import**, select **Import from first line** to import column names from the first nonblank line.

    The **Preview** panel updates to show the column headings from the first row.

8. Click **Next.**

    The **Column Attributes** panel shows the datatype, precision, scale, and format for each column.

9. Click **Next**.

    The **Name** field displays LA_Customers.

10. Select the **Tutorial_** project and the **Customers** folder.

11. Click **Finish**.

    The data object appears in the folder contents for the Customers folder.

Task 2. View the Data Object Properties

In this task, you can view the properties of the LA_Customers data object.

1. Click **Open** to open the **Library** workspace.

2. In the **Library** workspace, click **Data Objects** in the **Assets** panel.

    A list of data objects appears in the **Data Objects** panel.

3. Click **LA_Customers** flat file.

    The **Data Preview** panel appears with the data retrieved from the LA_Customers data object.

4. In the **Data Preview** panel, review the structure and content of the LA_Customers data object.

    The Analyst tool displays the first 100 rows of the flat file data object.

5. Click **Properties**.

    The **Properties** panel displays the name, type, description, and location of the data object. You can also see the column names and column properties for the data object.

Creating Data Objects Summary

In this lesson, you learned that data objects are representations of data based on a flat file or a relational database source. You learned that you can create a flat file data object and preview the data in it.

You uploaded a flat file and created a flat file data object, previewed the data for the data object, and viewed the properties for the data object.

After you create a data object, you create a default profile for the data object in Lesson 3, and you create a custom profile for the data object in Lesson 4.

CHAPTER 4

Lesson 3. Creating Default Profiles

This chapter includes the following topics:

Creating Default Profiles Overview

A profile is the analysis of data quality based on the content and structure of data. A default profile is a profile that you create with default options. Use a default profile to get profile results without configuring all columns and options for a profile.

Create and run a default profile to analyze the quality of the data when you start a data quality project. When you create a default profile object, you select the data object and the data object columns that you want to analyze. A default profile skips the profile column and option configuration. The Analyst tool performs profiling on the live flat file for the flat file data object.

Story

HypoStores wants to incorporate data from the newly-acquired Los Angeles office into its data warehouse. Before the data can be incorporated into the data warehouse, it needs to be cleansed. You are the analyst who is responsible for assessing the quality of the data and passing the information on to the developer who is responsible for cleansing the data. You want to view the profile results quickly and get a basic idea of the data quality.

Objectives

In this lesson, you complete the following tasks:

1. Create and run a default profile for the LA_Customers flat file data object.
2. View the profile results.

Prerequisites

Before you start this lesson, verify the following prerequisite:

- You have completed lessons 1 and 2 in this tutorial.

Set aside 5 to 10 minutes to complete this lesson.

Task 1. Create and Run a Default Profile

In this task, you create a default profile for all columns in the data object and use default sampling and drill-down options.

1. In the **Library** workspace, select a data object in the **Assets** panel.

2. Right-click the data object and select **Create Profile**.

   The **New Profile** wizard appears.

3. In the **Specify General Properties** screen, the name, description, and location is populated by default.

4. Click **Next**.

5. In the **Select Source** screen, the data object appears by default. You can view the columns in the **Choose Columns** pane.

6. Click **Next**.

7. In the **Specify Settings** screen, the following options are selected by default:

   - **Run column profile**

   - **All rows** in the **Run profile on** pane

   - **Live** in the **Drilldown** pane

   - **Exclude approved data types and data domains from the data type and data domain inference in the subsequent profile runs.**

   - **Native** option in the **Run-time environment** pane.

8. Click **Next**.

9. In the **Specify Rules and Filters** screen, click **Save and Run** to create and run the profile.

   The Analyst tool creates and runs the profile. The profile results appear in the summary view.

Task 2. View the Profile Results in Summary View

In this task, you use the summary view for the LA_Customers profile to get a quick overview of the profile results.

1. In the **Library** > **Assets** > **Profiles** pane, click the **LA_Customers** profile.

   The profile results appear in the summary view.

2. In the summary view, click **Columns** in the **Filter By** pane to view the profile results for columns.

3. Move the pointer over the horizontal bar charts to view the values in percentages.

4. In the **Null Distinct Non-Distinct %** section, you can view the null values, distinct values, and non-distinct values in percentages for a column.

5. In the **Pattern** section, you can view multiple patterns in the column as horizontal bar charts. You can view the pattern characters and the number of similar patterns in a column as a percentage when you move the pointer over the bar chart.

6. In the **Length** section, you can view the minimum and maximum length of the values in the column.

7. In the **Value** section, you can view the minimum and maximum values in a column.

8. In the **Data Type** section, you can view all the inferred data types and documented data types for a column when you move the pointer over the values.

9. In the **Data Domain** section, you can view all the inferred data domains for a column when you move the pointer over the values.

10. To view the outlier data, click **Actions** > **Detect Outlier** to detect outliers in the profile results.

11. Click **Pattern outlier** or **Value frequency outlier** filters to view the outliers in the profile results.

12. Click a column name to view the profile results for the column in the detailed view.

Creating Default Profiles Summary

In this lesson, you learned that a default profile shows profile results without configuring all columns and row sampling options for a profile. You learned that you create and run a default profile to analyze the quality of the data when you start a data quality project. You also learned that the Analyst tool performs profiling on the live flat file for the flat file data object.

You created a default profile and analyzed the profile results. You got more information about the columns in the profile, including null values and data types. You also used the column values and patterns to identify data quality issues.

After you analyze the results of a quick profile, you can complete the following tasks:

- Create a custom profile to exclude columns from the profile and only include the columns you are interested in.

- Create an expression rule to create virtual columns and profile them.

- Create a reference table to include valid values for a column.

CHAPTER 5

Lesson 4. Creating Custom Profiles

This chapter includes the following topics:

Creating Custom Profiles Overview

A profile is the analysis of data quality based on the content and structure of data. A custom profile is a profile that you create when you want to configure the columns, sampling options, and drilldown options for faster profiling. Configure sampling options to select the sample rows in the source. Configure drilldown options to drill down to data rows in the source data or staged data. You can choose to run the profile in a Hadoop or native environment.

You create and run a profile to analyze the quality of the data when you start a data quality project. When you create a profile object, you start by selecting the data object and data object columns that you want to run a profile on.

Story

HypoStores needs to incorporate data from the newly-acquired Los Angeles office into its data warehouse. HypoStores wants to access the quality of the customer tier data in the LA customer data file. You are the analyst responsible for assessing the quality of the data and passing the information on to the developer responsible for cleansing the data.

Objectives

In this lesson, you complete the following tasks:

1. Create a custom profile for the flat file data object and exclude the columns with null values.

2. Run the profile to analyze the content and structure of the CustomerTier column.

3. Drill down into the rows for the profile results.

Prerequisites

Before you start this lesson, verify the following prerequisite:

- You have completed lessons 1, 2, and 3 in this tutorial.

Timing

Set aside 5 to 10 minutes to complete this lesson.

Task 1. Create a Custom Profile

In this task, you create a custom profile. When you create a custom profile, you select the data object and the columns that you want to run a profile on. You also configure the sampling and drill-down options.

1. Click **New** > **Profile**.

   The **New Profile** wizard appears.

2. The **Single source** option is selected by default. Click **Next**.

3. In the **Specify General Properties** screen, set the following options:

   - In the Name field, enter **Profile_LA_Customers**.

   - In the Location field, select the **Customers** folder.

4. Click **Next**.

5. In the **Select Source** screen, click **Choose**.

   The **Choose Data Object** dialog box appears.

6. In the **Choose Data Object** dialog box, select **LA_Customers**. Click **OK**.

7. In the **Select Source** screen, clear the Address2, Address3, and City2 columns.

8. Click **Next**.

9. In the **Specify Settings** screen, set the following options:

   - **Run column profile**.

   - Select the **Random sample** option in the **Run profile on** pane.

   - Select **Exclude approved data types and data domains from the data type and data domain inference in the subsequent profile runs** option.

   - Select the **Staged** option in the **Drilldown** pane.

10. Click **Next**.

11. In the **Specify Rules and Filters** screen, click **Save and Finish** to create and run the profile.

    The Analyst tool creates the profile and displays the profile in the **Discovery** workspace. You need to run the profile to view the results.

Task 2. Run the Profile

In this task, you run a profile to perform profiling on the data object and display the profile results. The Analyst tool performs profiling on the staged flat file for the flat file data object.

1. Verify that you are in the **Discovery** workspace.

   You can see the profile **Profile_LA_Customers_Custom** in the workspace.

2. Click **Profile_LA_Customers_Custom** in the workspace.

3. The profile screen appears where you can choose to edit the profile or run the profile. Click **Run**.

   The profile results appear in the summary view.

Task 3. Drill Down on Profile Results

In this task, you drill down on the CustomerTier column values to see the source rows in the data object for the profile.

1. Verify that you are in the summary view of the profile results for the **Profile_LA_Customers** profile.

2. Click the **CustomerTier** column.

   The profile results for the column appear in the detailed view.

3. In the detailed view, select the Diamond, Ruby, Emerald, and Bronze values. Right-click on the values in the **Values** pane, and select **Drilldown**.

The rows for the column with a value of Diamond, Ruby, Emerald, or Bronze appear in the **Data Preview** pane.

The following image shows the drill-down results in the **Data Preview** pane when you drill down on values Diamond, Ruby, Emerald, or Bronze:
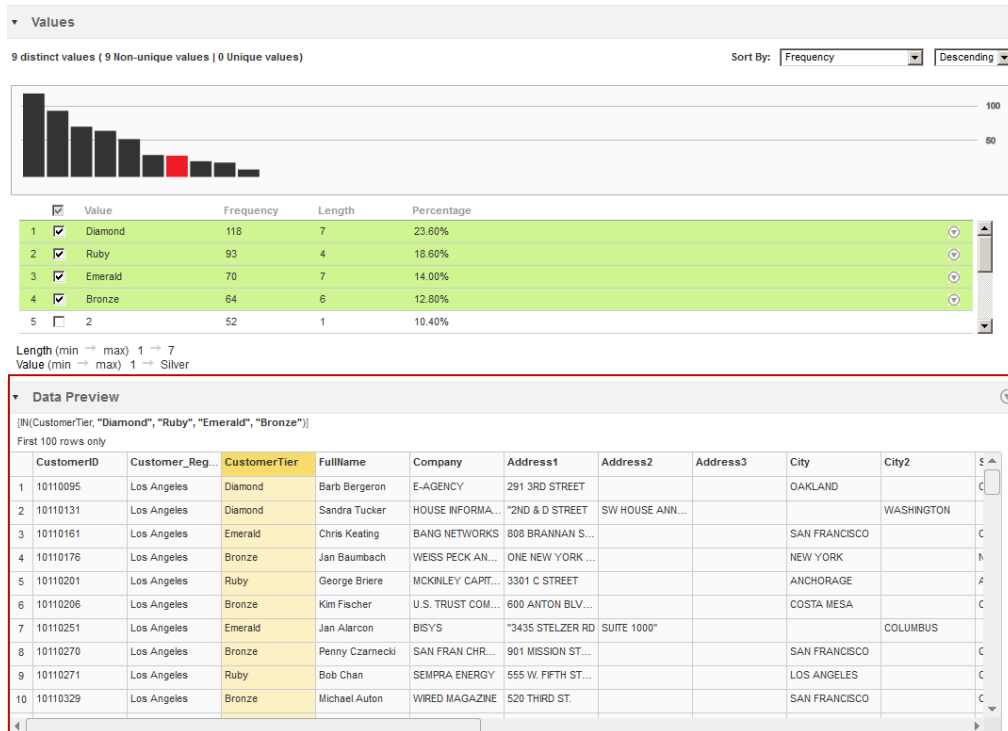
The **Data Preview** pane displays the first 100 rows for the selected column. The title of the **Data Preview** pane shows the logic used for the source column.

Creating Custom Profiles Summary

In this lesson, you learned that you can configure the columns that get profiled and that you can configure the sampling and drilldown options. You learned that you can drill down to see the underlying rows for column values and that you can configure the columns that are included when you view the column values.

You created a custom profile that included the CustomerTier column, ran the profile, and drilled down to the underlying rows for the CustomerTier column in the results.

Use the custom profile object to create an expression rule in lesson 5.

CHAPTER 6

Lesson 5. Creating Expression Rules

This chapter includes the following topics:

Creating Expression Rules Overview

Expression rules use expression functions and source columns to define rule logic. You can create expression rules and add them to a profile in the Analyst tool. An expression rule can be associated with one or more profiles.

The output of an expression rule is a virtual column in the profile. The Analyst tool profiles the virtual column when you run the profile.

You can use expression rules to validate source columns or create additional source columns based on the value of the source columns.

Story

HypoStores wants to incorporate data from the newly-acquired Los Angeles office into its data warehouse. HypoStores wants to analyze the customer names and separate customer names into first name and last name. HypoStores wants to use expression rules to parse a column that contains first and last names into separate virtual columns and then profile the columns. HypoStores also wants to make the rules available to other analysts who need to analyze the output of these rules.

Objectives

In this lesson, you complete the following tasks:

1. Create expression rules to separate the FullName column into first name and last name columns. You create a rule that separates the first name from the full name. You create another rule that separates the last name from the first name. You create these rules for the Profile_LA_Customers profile.

2. Run the profile and view the output of the rules in the profile.

3. Edit the rules to make them usable for other Analyst tool users.

Prerequisites

Before you start this lesson, verify the following prerequisite:

- You have completed Lessons 1 through 4.

Timing

Set aside 10 to 15 minutes to complete this lesson.

Task 1. Create Expression Rules and Run the Profile

In this task, you create two expression rules to parse the FullName column into two virtual columns named FirstName and LastName. The rule names are FirstName and LastName.

1. In the **Library** workspace, click the **Profile_LA_Customers** profile.

    The profile results appear in summary view.

2. Click **Edit**.

    The **Profile** wizard appears.

3. Click **Specify Rules and Filters** screen.

4. In the **Rules** pane, click **Actions** > **Create Rule**.

5. In the **Name** field, enter **FirstName**.

6. In the **Expression** section, enter the following expression to separate the first name from the **Name** column:

    ```
    SUBSTR(FullName,1,INSTR(FullName,' ' ,-1,1 ) - 1)
    ```

7. Click **Validate**.

8. Click **OK**.

9. Repeat steps 4 through 8 to create a rule named **LastName**. Enter the following expression to separate the last name from the **Name** column:

    ```
    SUBSTR(FullName,INSTR(FullName,' ',-1,1),LENGTH(FullName))
    ```

10. Click **Save and Run** to save and run the profile.

Task 2. View the Expression Rule Output

In this task, you view the output of expression rules that separates first and last names after you run a profile.

1. In the summary view, click **Edit**.

    The profile wizard appears

2. In the profile wizard, click **Select Source**.

3. In the **Select Source** screen, select the check box next to **Name** on the toolbar to clear all columns.

    One of the columns is selected by default because you need to select at least one column in the **Columns** section.

4. Select the **FullName** column and the **FirstName** and **LastName** rules.

5. Clear any other column that is selected.

6. Click **Save and Run**.

   The profile results appear in the summary view.

7. Click the **FirstName** rule, the profile results for the rule appears in detailed view.

8. Select a value in the **Values** pane. Right-click on the value and click **Drilldown**.

   The values for the **FullName** column and the **FirstName** and **LastName** rules appear in the **Data Preview** pane along with other column values. Notice that the Analyst tool separates the **FullName** column into first name and last name.

Task 3. Edit the Expression Rules

In this task, you make the expression rules reusable and available to all Analyst tool users.

1. In the summary view for the **Profile_LA_Customers** profile, click **Edit**.

   The profile wizard appears.

2. Click the **Specify Rules and Filters** screen.

3. In the **Specify Rules and Filters** screen, select the **FirstName** rule and click **Actions** > **Edit Rule**.

   The **Edit Rule** dialog box appears.

4. Select the **Do you want to save this rule as a reusable rule?** option, and choose a location to save the rule.

5. Click **OK**.

6. Select the **LastName** rule, and repeat steps 3 through 5.

7. Click **Save and Finish** to save the profile.

   Any Analyst tool user can use the FirstName and LastName rules to split a column with first and last names into separate columns.

Creating Expression Rules Summary

In this lesson, you learned that expression rules use expression functions and source columns to define rule logic. You learned that the output of an expression rule is a virtual column in the profile. The Analyst tool includes the virtual column when you run the profile.

You created two expression rules, added them to a profile, and ran the profile. You viewed the output of the rules and made them available to all Analyst tool users.

CHAPTER 7

Lesson 6. Creating and Running Scorecards

This chapter includes the following topics:

Creating and Running Scorecards Overview

A scorecard is the graphical representation of valid values for a column or the output of a rule in profile results. Use scorecards to measure and monitor data quality progress over time.

To create a scorecard, you add columns from the profile to a scorecard as metrics, assign weights to metrics, and configure the score thresholds. You can add filters to the scorecards based on the source data. To run a scorecard, you select the valid values for the metric and run the scorecard to see the scores for the metrics.

Scorecards display the value frequency for columns in a profile as scores. Scores reflect the percentage of valid values for a metric.

Story

HypoStores wants to incorporate data from the newly-acquired Los Angeles office into its data warehouse. Before the organization merges the data, they want to verify that the data in different customer tiers and states is analyzed for data quality. You are the analyst who is responsible for monitoring the progress of performing the data quality analysis. You want to create a scorecard from the customer tier and state profile columns, configure thresholds for data quality, and view the score trend charts to determine how the scores improve over time.

34

Objectives

In this lesson, you will complete the following tasks:

1. Create a scorecard from the results of the Profile_LA_Customers_Custom profile to view the scores for the CustomerTier and State columns.
2. Run the scorecard to generate the scores for the CustomerTier and State columns.
3. View the scorecard to see the scores for each column.
4. Edit the scorecard to specify different valid values for the scores.
5. Configure score thresholds, and run the scorecard.
6. View score trend charts to determine how scores improve over time.

Prerequisites

Before you start this lesson, verify the following prerequisite:

- You have completed lessons 1 through 5 in this tutorial.

Timing

Set aside 15 minutes to complete the tasks in this lesson.

Task 1. Create a Scorecard from the Profile Results

In this task, you create a scorecard from the Profile_LA_Customers_Custom profile to score the CustomerTier and State column values.

1. In the **Library** workspace, click the **Profile_LA_Customers** profile.

   The summary view of the profile results appear.

2. In the summary view, select the **CustomerTier** column, and right-click on the column, and select **Add to >
   Scorecard**.

   The **Add to Scorecard** wizard appears.

3. In the **Add to Scorecard** wizard, the **New Scorecard** option is selected by default. Click **Next**.

4. In the **Step 2 of 8** screen, enter **sc_LA_Customer** for the scorecard name, and navigate to the **Customers**
   folder for the scorecard location.

5. Click **Next**.

6. In the **Step 3 of 8** screen, select the **CustomerTier** and **State** columns to add to the scorecard.

7. Click **Next**.

8. In the **Step 4 of 8** screen, you can create, edit, or delete filters for the metrics. In this tutorial, we will not
   create a scorecard filter. Click **Next**.

9. In the **Step 5 of 8** screen, select the **CustomerTier** metric in the **Metrics** pane.

10. In the **Score using: Values** pane, select all the values, and click the **Add All** button to move the values to
    the **Valid Values** section.

    Use the **Shift** key to select multiple values.

11. In the **Metrics** pane, select the **State** metric, and select those values that have two letter state codes in
    the **Score using: Values** section.

12. Click the **Add** button to move the values to the **Valid Values** section.

You can see the total number of valid values and valid value percentage at the top of the section.

13. For each metric in the **Metrics** section, accept the default settings for the score thresholds in the **Metric Thresholds** section.

14. Click **Next**.

15. In the **Step 6 of 8** screen, you can optionally select a metric group to add the metrics. By default, the Analyst tool adds the metrics to the **Default** metric group.

16. Click **Next**.

17. In the **Step 7 of 8** screen, double-click the **Weight** column for the **CustomerTier** metric in the **Default - Metrics** pane.

    When you run a scorecard, the Analyst tool calculates the weighted average for each metric group based on the metric score and weight you assign to each metric.

18. Enter a weight for the **CustomerTier** and **State** metrics.

19. Click **Next**.

20. In the **Step 8 of 8** screen, the **Native** option is selected by default. Click **Save** to create the scorecard.

    The scorecard appears in the **Scorecards** workspace.

Task 2. Run the Scorecard

In this task, you run the sc_LA_Customer scorecard to generate the scores for the CustomerTier and State columns.

1. In the **Library** workspace, click **Assets** > **Scorecards**.

2. Select a scorecard in the **Scorecards** pane.

3. Click **Actions** > **Open**.

    The scorecard appears.

4. Click **Actions** > **Run Scorecard**

    The **Run Scorecard** dialog box appears.

5. Verify the settings in the dialog box, and click **Run**.

    The **Scorecards** workspace displays the scores for the CustomerTier and State columns.

Task 3. View the Scorecard

In this task, you view the sc_LA_Customer scorecard to see the scores for the CustomerTier and State columns.

1. Select the **State** row that contains the State score you want to view.

    In the **sc_LA_Customer - metrics** section, you can view the following properties of the scorecard:

    - Scorecard name.
    - Total number of rows in the scorecard.

- Number of rows that are not valid.

- Score along with a horizontal bar chart.

- Score trend. You can click on the score trend to view a graphical representation in the **Trend Chart Detail** screen.

- Weight of the metric.

- Cost of invalid data.

- Cost trend.

- Data object. Click the data object to view the data preview of the data object in the **Discovery** workspace.

- Column or rule name.

- Type of source.

- Drilldown icon.

2. Click the drilldown icon in the State row.

   The scores that are not valid for the State column appears in the **Invalid Rows** section in the **Drilldown** pane.

3. Select **Valid Rows** to view the scores that are valid for the State column.

4. Click the drilldown icon in the **CustomerTier** row.

   All scores for the **CustomerTier** column are valid.

Task 4. Edit the Scorecard

In this task, you edit the sc_LA_Customer scorecard to specify the Ruby value as not valid for the CustomerTier score.

1. Verify the you are in the **Scorecard** workspace, and the sc_LA_Customer scorecard is open.

2. Select **Actions** > **Edit** > **Metrics**.

   The **Edit Scorecard** dialog box appears.

3. In the **Metrics** section, select **CustomerTier**.

4. In the **Score using: Values** section, move **Ruby** from the **Valid Values** section to the **Available Values** section.

   Accept the default settings in the **Metric Thresholds** section.

5. Click **Save & Run** to save the changes to the scorecard and run it.

6. View the CustomerTier score again.

   The CustomerTier score changes to 81.4 percentage.

Task 5. Configure Thresholds

In this task, you configure thresholds for the State score in the sc_LA_Customer scorecard to determine the acceptable ranges for the data in the State column. Values with a two letter code, such as CA are acceptable, and codes with more than two letters such as Calif are not acceptable.

1. Verify the you are in the **Scorecard** workspace, and the sc_LA_Customer scorecard is open.

2. Select **Actions** > **Edit** > **Metrics**.

   The **Edit Scorecard** dialog box appears.

3. In the **Metrics** section, select **State**.

4. In the **Metric Thresholds** section, enter the following ranges for the Good and Unacceptable scores: 90 to 100% Good; 0 to 50% Unacceptable; 51% to 89% Acceptable.

   The thresholds represent the lower bounds of the acceptable and good ranges.

5. Click **Save & Run** to save the changes to the scorecard and run it.

   In the **Scorecard** panel, view the changes to the score percentage and the score displayed as a bar for the State score.

Task 6. View Score Trend Charts

In this task, you view the trend chart for the State score. You can view trend charts to monitor scores over time.

1. Verify the you are in the **Scorecard** workspace, and the sc_LA_Customer scorecard is open.

2. Select **State** row.

3. Click **Actions** > **Show Trend Chart**, or click the arrow under the **Score Trend** column.

   The **Trend Chart Detail** dialog box appears. You can view the **Good**, **Acceptable**, and **Unacceptable** thresholds for the score. The thresholds change each time you run the scorecard after editing the values for scores in the scorecard.

4. Point to any circle in the chart to view the valid values in the **Valid Values** section at the bottom of the chart.

5. Click **Close** to return to the scorecard.

Creating and Running Scorecards Summary

In this lesson, you learned that you can create a scorecard from the results of a profile. A scorecard contains the columns from a profile. You learned that you can run a scorecard to generate scores for columns. You edited a scorecard to configure valid values and set thresholds for scores. You also learned how to view the score trend chart.

You created a scorecard from the CustomerTier and State columns in a profile to analyze data quality for the customer tier and state columns. You ran the scorecard to generate scores for each column. You edited the scorecard to specify different valid values for scores. You configured thresholds for a score and viewed the score trend chart.

CHAPTER 8

Lesson 7. Creating Reference Tables from Profile Columns

This chapter includes the following topics:

Creating Reference Tables from Profile Columns Overview

A reference table contains reference data that you can use to standardize source data. Reference data can include valid and standard values. Create reference tables to establish relationships between source data values and the valid and standard values.

You can create a reference table from the results of a profile. After you create a reference table, you can edit the reference table to add columns or rows and add or edit standard and valid values. You can view the changes made to a reference table in an audit trail.

Story

HypoStores wants to profile the data to uncover anomalies and standardize the data with valid values. You are the analyst who is responsible for standardizing the valid values in the data. You want to create a reference table based on valid values from profile columns.

Objectives

In this lesson, you complete the following tasks:

1. Create a reference table from the CustomerTier column in the Profile_LA_Customers_Custom profile by selecting valid values for columns.
2. Edit the reference table to configure different valid values for columns.

Prerequisites

Before you start this lesson, verify the following prerequisite:

- You have completed lessons 1 through 6 in this tutorial.

Set aside 15 minutes to complete the tasks in this lesson.

Task 1. Create a Reference Table from Profile Columns

In this task, you create a reference table and add the CustomerTier column from the Profile_LA_Customers profile to the reference table.

1.   In the **Library** workspace, click **Assets** > **Profiles**.

2.   Click **Profile_LA_Customers** profile to open the profile results in summary view.

3.   In the summary view, select the CustomerTier column that you want to add to the reference table. Right-click and select **Add to Reference Table**.

     The **Add to Reference Table** dialog box appears.

4.   Select **Create a reference table**.

5.   Click **Next**.

6.   In the **Name** field, enter **Reftab_CustTier_HypoStores**.

7.   Enter a description and set 0 as the default value.

     The Analyst tool uses the default value for any table record that does not contain a value.

8.   Click **Next**.

9.   In the **Column Attributes** section, configure the following column properties for the CustomerTier column:

Property	Description
Name	CustomerTier
Data type	String
Precision	10
Description	Reference customer tier values

10. Optionally, choose to create a description column for rows in the reference table. Enter the name and precision for the column.

11. Verify the CustomerTier column values in the **Preview** section.

12. Click **Next**.

     The Reftab_CustomerTier_HypoStores reference table name appears. You can enter an optional description.

13. In the **Save in** section, select your tutorial project where you want to create the reference table.

     The **Reference Tables:** panel lists the reference tables in the location you select.

14. Enter an optional audit note.

15. Click **Finish**.

Task 2. Edit the Reference Table

In this task, you edit the Reftab_CustomerTier_HypoStores table to add alternate values for the customer tiers.

1. In the **Library** workspace, click **Assets** > **Reference Tables**.
2. Click the Reftab_CustomerTier_HypoStores reference table.

    The reference table opens in the **Design** workspace.
3. To edit a row, select the row and click **Actions** > **Edit** or click the **Edit** icon.

    The **Edit Row** dialog box appears. Optionally, select multiple rows to add the same alternate value to each row.
4. Enter the following alternate values for the Diamond, Emerald, Gold, Silver, and Bronze rows: 1, 2, 3, 4, 5.

    Enter an optional audit note.
5. Click **Apply** to apply the changes.
6. Click **Close**.

    The changed reference table values appear in the **Design** workspace.

Creating Reference Tables from Profile Columns Summary

In this lesson, you learned how to create reference tables from the results of a profile to configure valid values for source data.

You created a reference table from a profile column by selecting valid values for columns. You edited the reference table to configure different valid values for columns.

CHAPTER 9

Lesson 8. Creating Reference Tables

This chapter includes the following topics:

Creating Reference Tables Overview

A reference table contains reference data that you can use to standardize source data. Reference data can include valid and standard values. Create reference tables to establish relationships between the source data values and the valid and standard values.

You can manually create a reference table using the reference table editor. Use the reference table to define and standardize the source data. You can share the reference table with a developer to use in Standardizer and Lookup transformations in the Developer tool.

Story

HypoStores wants to standardize data with valid values. You are the analyst who is responsible for standardizing the valid values in the data. You want to create a reference table to define standard customer tier codes that reference the LA customer data. You can then share the reference table with a developer.

Objectives

In this lesson, you complete the following task:

- Create a reference table using the reference table editor to define standard customer tier codes that reference the LA customer data.

Prerequisites

Before you start this lesson, verify the following prerequisite:

- You have completed lessons 1 and 2 in this tutorial.

Timing

Set aside 10 minutes to complete the task in this lesson.

Task 1. Create a Reference Table

In this task, you will create the Reftab_CustomerTier_Codes reference table to standardize the valid values for the customer tier data.

1. Click **New** > **Reference Table**.

    The **New Reference Table** wizard appears.

2. Select **Use the reference table editor**.

3. Click **Next**.

4. For each column you want to include in the reference table, click the **Add New Column** icon and configure the column properties for each column.

    Add the following column names: CustomerID, CustomerTier, and Status. You can reorder the columns or delete columns.

5. Enter an optional description and set the default value to **0**.

    The Analyst tool uses the default value for any table record that does not contain a value.

6. Click **Next**.

7. In the **Name** field, enter **Reftab_CustomerTier_Codes**.

8. In the **Folders** section, select the **Customers** folder in the tutorial project.

9. Click **Finish**.

    The reference table appears in the **Design** workspace.

10. From the **Actions** menu, select **Add Row** to populate each reference table column with the following four values:

    CustomerID = LA1, LA2, LA3, LA4

    CustomerTier = 1, 2, 3, 4.

    Status= Active, Inactive

Creating Reference Tables Summary

In this lesson, you learned how to create reference tables using the reference table editor to create standard valid values to use with source data.

You created a reference table using the reference table editor to standardize the customer tier values for the LA customer data.

Part II: Getting Started with Informatica Developer

This part contains the following chapters:

CHAPTER 10

Lesson 1. Setting Up Informatica Developer

This chapter includes the following topics:

Setting Up Informatica Developer Overview

Before you start the lessons in this tutorial, you must start and set up the Developer tool. To set up the Developer tool, you add a domain. You add a Model repository that is in the domain, and you create a project and folder to store your work. You must also select a default Data Integration Service if the domain includes more than one service.

The Informatica domain is a collection of nodes and services that define the Informatica environment. Services in the domain include the Model Repository Service and the Data Integration Service.

The Model Repository Service manages the Model repository. The Model repository is a relational database that stores the metadata for projects that you create in the Developer tool. A project stores objects that you create in the Developer tool. A project can also contain folders that store related objects, such as objects that are part of the same business requirement.

The Data Integration Service performs data integration tasks in the Developer tool.

Objectives

In this lesson, you complete the following tasks:

- Start the Developer tool and go to the Developer tool workbench.
- Add a domain in the Developer tool.
- Add a Model repository so that you can create a project.

- Create a project to store the objects that you create in the Developer tool.
- Create a folder in the project that can store related objects.
- Select a default Data Integration Service to perform data integration tasks.

Prerequisites

Before you start this lesson, verify the following prerequisites:

- You have installed the Developer tool.
- You have a domain name, host name, and port number to connect to a domain. You can get this information from a domain administrator.
- A domain administrator has configured a Model Repository Service in the Administrator tool.
- You have a user name and password to access the Model Repository Service. You can get this information from a domain administrator.
- A domain administrator has configured a Data Integration Service.
- The Data Integration Service is running.

Timing

Set aside 5 to 10 minutes to complete the tasks in this lesson.

Task 1. Start Informatica Developer

Start the Developer tool to begin the tutorial.

1. Start the Developer tool.

    The **Welcome** page of the Developer tool appears.

2. Click the **Workbench** button.

    The Developer tool workbench appears.

Task 2. Add a Domain

In this task, you add a domain in the Developer tool to access a Model repository.

1. Click **Window** > **Preferences**.

    The **Preferences** dialog box appears.

2. Select **Informatica** > **Domains**.

3. Click **Add**.

    The **New Domain** dialog box appears.

4. Enter the domain name, host name, and port number.

5. Click **Finish**.

6. Click **OK**.

Task 3. Add a Model Repository

In this task, you add the Model repository that you want to use to store projects and folders.

1. Click **File** > **Connect to Repository**.

   The **Connect to Repository** dialog box appears.

2. Click **Browse** to select a Model Repository Service.

3. Click **OK**.

4. Click **Next**.

5. Enter your user name and password.

6. Select a namespace.

7. Click **Finish**.

   The Model repository appears in the **Object Explorer** view.

Task 4. Create a Project

In this task, you create a project to store objects that you create in the Developer tool. You can create one project for all tutorials in this guide.

1. In the **Object Explorer** view, select a Model Repository Service.

2. Click **File** > **New** > **Project**.

   The **New Project** dialog box appears.

3. Enter your name prefixed by "Tutorial_" as the name of the project.

4. Click **Finish**.

   The project appears under the Model Repository Service in the **Object Explorer** view.

Task 5. Create a Folder

In this task, you create a folder to store related objects. You can create one folder for all tutorials in this guide.

1. In the **Object Explorer** view, select the project that you want to add the folder to.

2. Click **File** > **New** > **Folder**.

3. Enter a name for the folder.

4. Click **Finish**.

   The Developer tool adds the folder under the project in the **Object Explorer** view. Expand the project to see the folder.

Task 6. Select a Default Data Integration Service

In this task, you select a default Data Integration Service so you can run mappings and preview data. This step is required if there is more than one Data Integration Service in the domain. If the domain contains one Data Integration Service, this service is set as the default.

1. Click **Window** > **Preferences**.

   The **Preferences** dialog box appears.

2. Select **Informatica** > **Data Integration Services**.

3. Expand the domain.

4. Select a Data Integration Service.

5. Click **Set as Default**.

6. Click **OK**.

Setting Up Informatica Developer Summary

In this lesson, you learned that the Informatica domain includes the Model Repository Service and Data Integration Service. The Model Repository Service manages the Model repository. A Model repository contains projects and folders. The Data Integration Service performs data integration tasks.

You started the Developer tool and set up the Developer tool. You added a domain to the Developer tool, added a Model repository, and created a project and folder. You also selected a default Data Integration Service.

Now, you can use the Developer tool to complete other lessons in this tutorial.

CHAPTER 11

Lesson 2: Importing Physical Data Objects

This chapter includes the following topics:

Importing Physical Data Objects Overview

A physical data object is a representation of data based on a flat file or relational database table. You can import a flat file or relational database table as a physical data object to use as a source or target in a mapping.

Story

HypoStores Corporation stores customer data from the Los Angeles office and Boston office in flat files. You want to work with this customer data in the Developer tool. To do this, you need to import each flat file as a physical data object.

Objectives

In this lesson, you import flat files as physical data objects. You also set the source file directory so that the Data Integration Service can read the source data from the correct directory.

Prerequisites

Before you start this lesson, verify the following prerequisite:

- You have completed lesson 1 in this tutorial.

Timing

Set aside 10 to 15 minutes to complete the tasks in this lesson.

Task 1. Import the Boston_Customers Flat File Data Object

In this task, you import a physical data object from a file that contains customer data from the Boston office.

1. In the **Object Explorer** view, select the Tutorial_Objects folder.
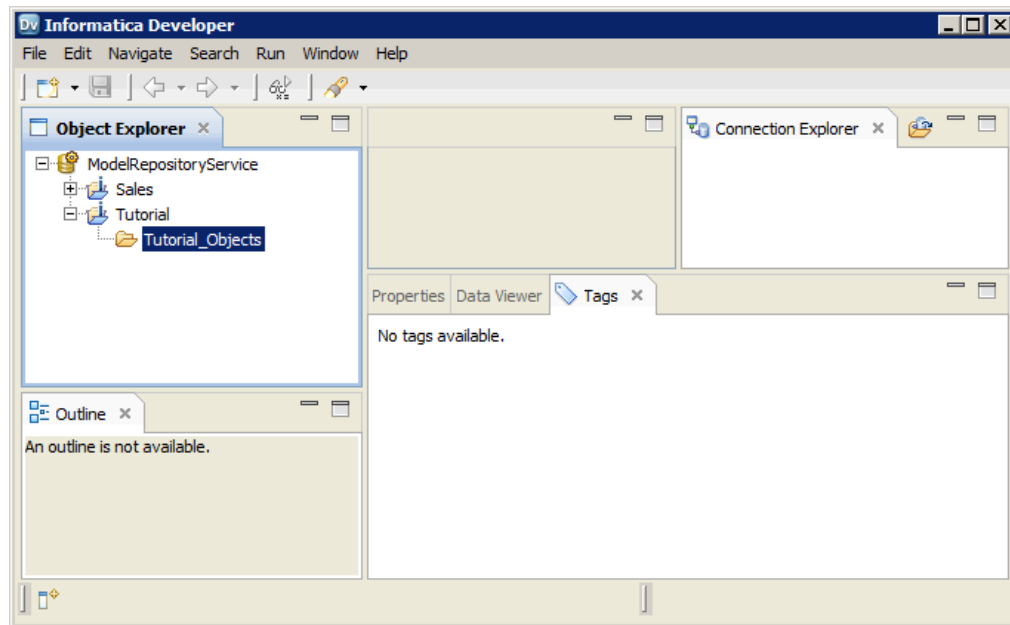
2. Right-click the Tutorial_Objects folder and select **New** > **Data Object**.

The **New** dialog box appears.
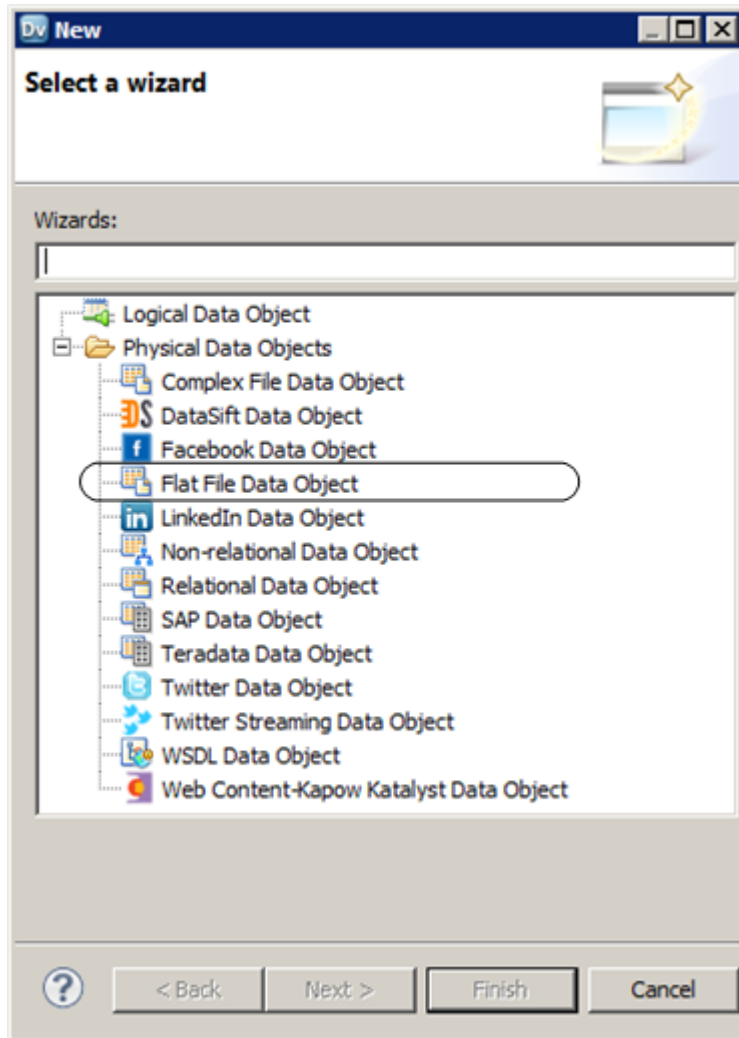
3. Select **Physical Data Objects** > **Flat File Data Object** and click **Next**.

The **New Flat File Data Object** dialog box appears.
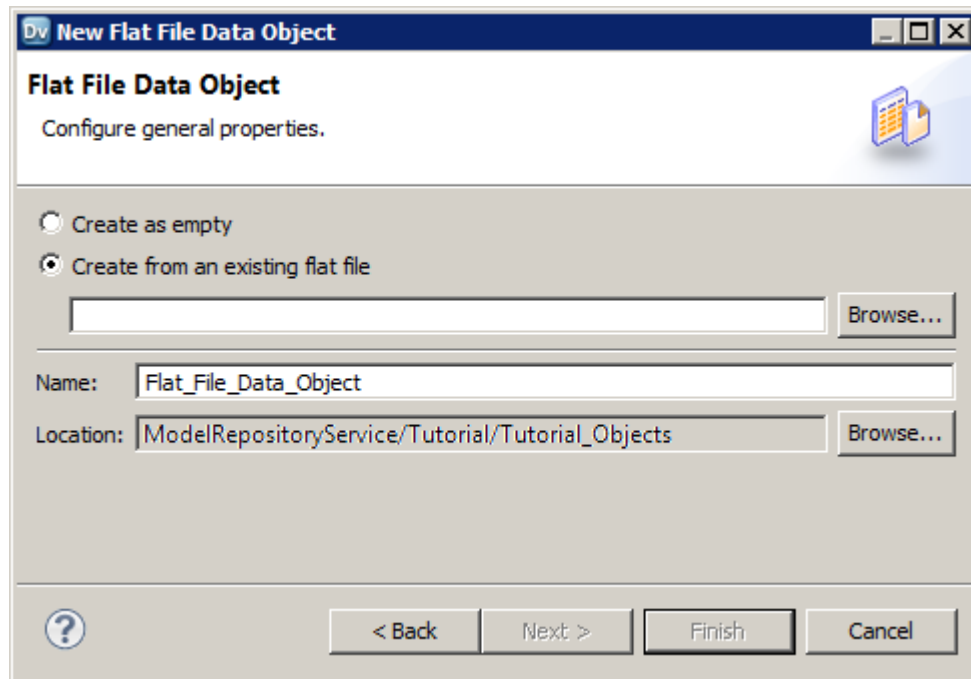
4. Select **Create from an existing flat file**.

5. Click **Browse** and navigate to Boston_Customers.csv in the following directory on the Developer tool machine: `<Informatica installation directory>\clients\DeveloperClient\Tutorials`

6. Click **Open**.

    The wizard names the data object "Boston_Customers."

7. Click **Next**.

8. Verify that the code page is set to **MS Windows Latin 1 (ANSI), superset of Latin 1** and the format is set to **Delimited**.

The **New Flat File Data Object** dialog box shows the default code page, the format, and a preview of the flat file data.
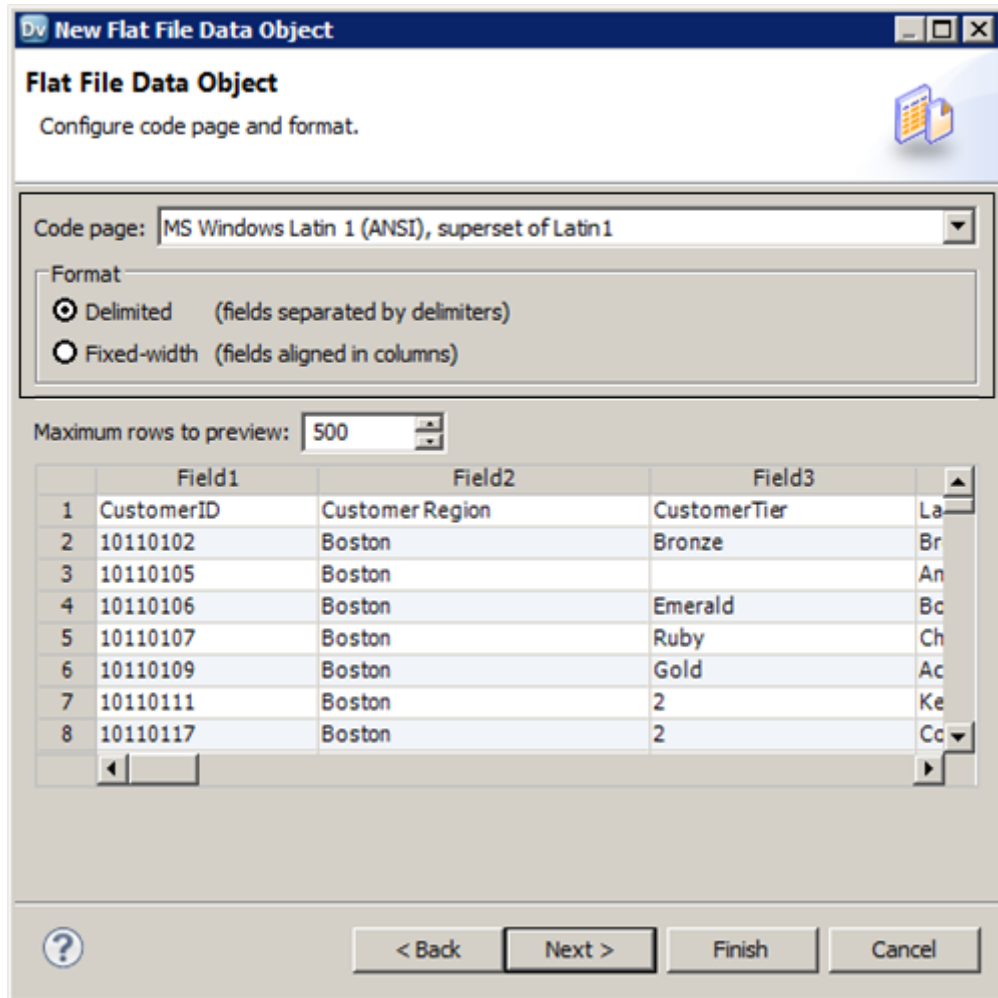
9. Click **Next**.
10. Select **Import column names from first line**.

The **New Flat File Data Object** dialog box shows the column names in the preview of the flat file data.
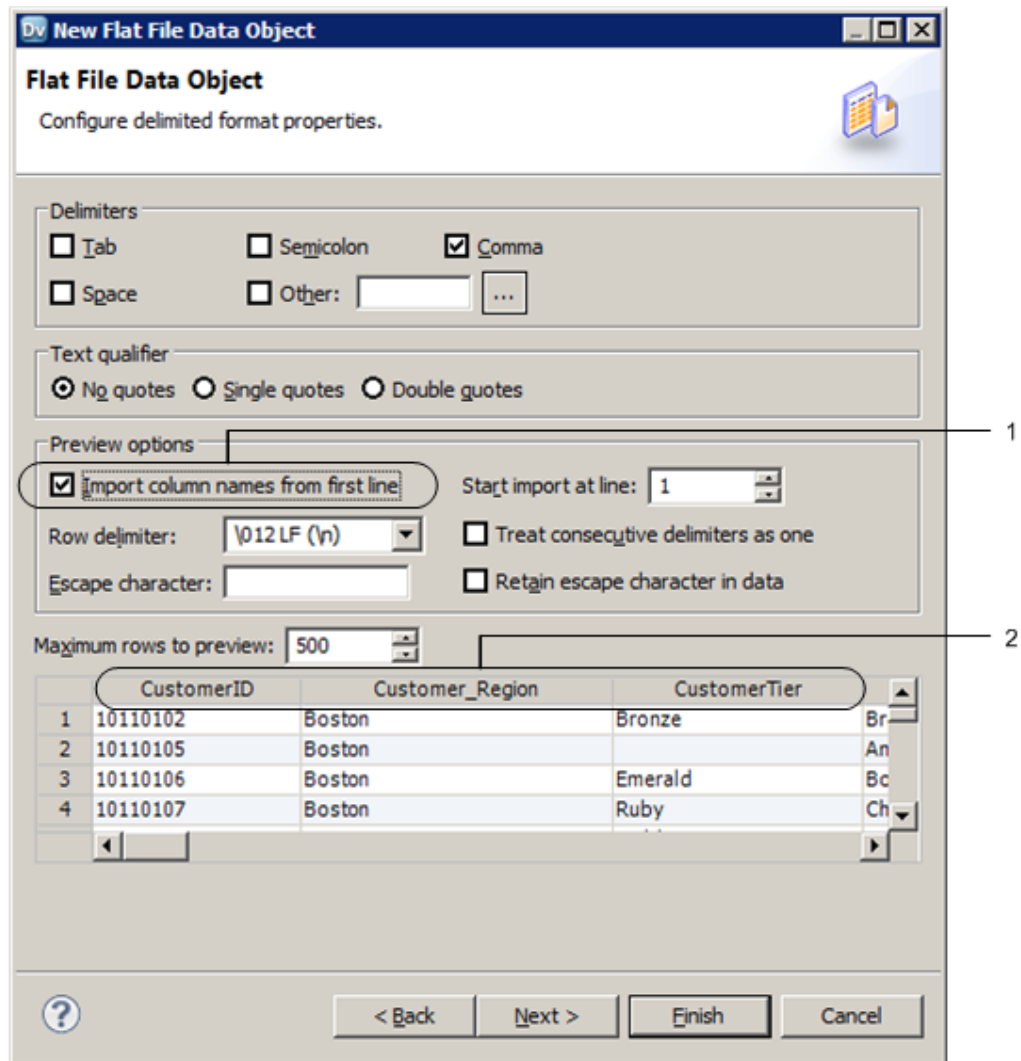
1. The **Import column names from first line** option
2. Column names

11. Click **Finish**.

The Boston_Customers physical data object appears under the Physical Data Objects folder in the Tutorial_Objects folder. The **Overview** view displays the file content and it is open in the editor.
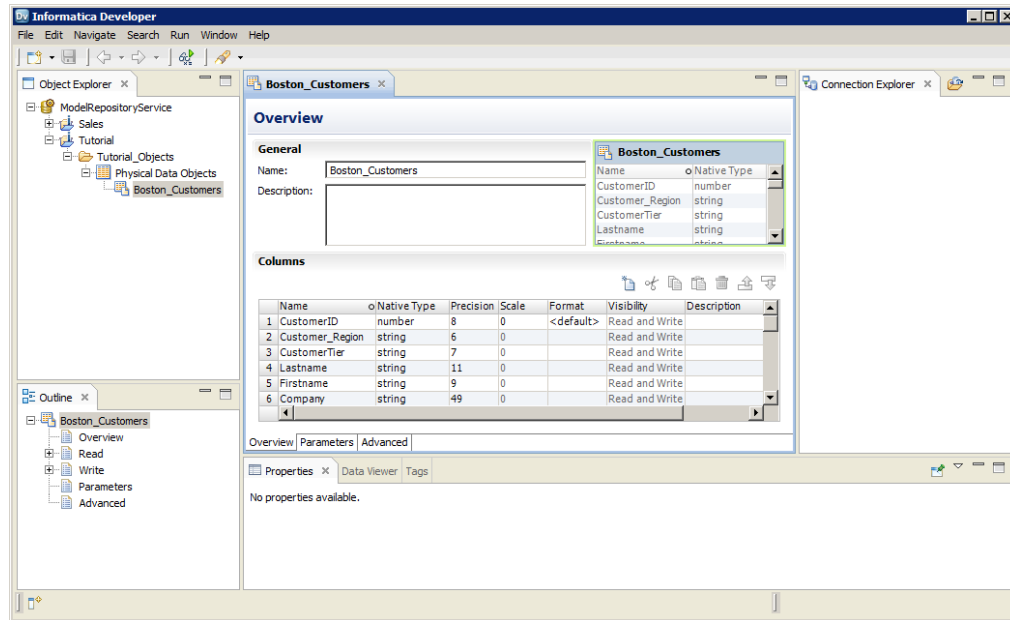
12. Click the **Advanced** view.

    The **Advanced** view shows properties for the physical data object.

13. In the **Advanced** view, scroll to the **Run-time: Read** section.

14. In the **Run-time: Read** section, set **Source file directory** to the following directory on the Data Integration Service machine: `<Informatica installation directory>\server\Tutorials`

    The Data Integration Service searches for the source file in the server directory on the machine that runs the Data Integration Service. The server installation contains a copy of the tutorial files. The Data Integration Service cannot read files from the client installation directory unless you change access permissions on the source file and directory.
    The following figure shows a sample source file directory:
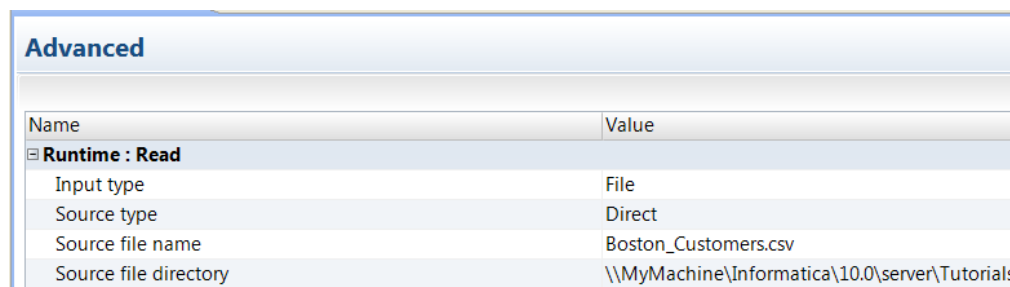
    Note: The Developer tool machine must have access to the source file directory on the machine that runs the Data Integration Service. If the Developer tool cannot access the source file directory, the Developer tool cannot preview data in the source file or run mappings that access data in the source file. If you run multiple Data Integration Services, there is a separate source file directory for each Data Integration Service.

15. Click the **Data Viewer** view.

16. In the **Data Viewer** view, click **Run**.

The Data Integration Service reads the data from the Boston_Customers file and shows the results in the Output window.

17. Click **File** > **Save** to save the Boston_Customers physical data object.

Task 2. Import the LA_Customers Flat File Data Object

In this task, you import a physical data object from a flat file that contains customer data from the Los Angeles office.

1. In the **Object Explorer** view, select the tutorial project.

2. Click **File** > **New** > **Data Object**.

   The **New** dialog box appears.

3. Select **Physical Data Objects** > **Flat File Data Object** and click **Next**.

   The **New Flat File Data Object** dialog box appears.

4. Select **Create from an Existing Flat File**.

5. Click **Browse** and navigate to LA_Customers.csv in the following directory: `<Informatica Installation Directory>\clients\DeveloperClient\Tutorials`

6. Click **Open**.

   The wizard names the data object LA_Customers.

7. Click **Next**.

8. Verify that the code page is MS Windows Latin 1 (ANSI), superset of Latin 1.

9. Verify that the format is delimited.

10. Click **Next**.

11. Verify that the delimiter is set to comma.

12. Select **Import column names from first line**.

13. Click **Finish**.

    The LA_Customers physical data object appears under **Physical Data Objects** in the tutorial project.

14. Click the **Read** view and select the Output transformation.

15. Click the **Runtime** tab on the **Properties** view.

16. Set the Source File Directory to the following directory on the Data Integration Service machine: `<Informatica Installation Directory>\server\Tutorials`

17. Click **File** > **Save**.

Task 3. Importing the All_Customers Flat File Data Object

In this task, you import a physical data object from a flat file that combines the customer order data from the Los Angeles and Boston offices.

1. In the **Object Explorer** view, select the tutorial project.
2. Click **File** > **New** > **Data Object**.

   The **New** dialog box appears.
3. Select **Physical Data Objects** > **Flat File Data Object** and click **Next**.

   The **New Flat File Data Source** dialog box appears.
4. Select **Create from an Existing Flat File**.
5. Click **Browse** and navigate to All_Customers.csv in the following directory: `<Informatica Installation Directory>\clients\DeveloperClient\Tutorials`.
6. Click **Open**.

   The wizard names the data object All_Customers.
7. Click **Next**.
8. Verify that the code page is MS Windows Latin 1 (ANSI), superset of Latin 1.
9. Verify that the format is delimited.
10. Click **Next**.
11. Verify that the delimiter is set to comma.
12. Select **Import column names from first line**.
13. Click **Finish**.

    The All_Customers physical data object appears under **Physical Data Objects** in the tutorial project.
14. Click the **Read** view and select the Output transformation.
15. Click the **Runtime** tab on the **Properties** view.
16. Set the Source File Directory to the following directory on the Data Integration Service machine:
    `<Informatica Installation Directory>\server\Tutorials`
17. Click **File** > **Save**.

Importing Physical Data Objects Summary

In this lesson, you learned that physical data objects are representations of data based on a flat file or a relational database table.

You created physical data objects from flat files. You also set the source file directory so that the Data Integration Service can read the source data from the correct directory.

You use the data objects as mapping sources in the data quality lessons.

Lesson 3. Run a Profile on Source Data

This chapter includes the following topics:

Profiling Data Overview

A profile is a set of metadata that describes the content and structure of a data set.

Profiling and data discovery is often the first step in a project. You can run a profile to evaluate the structure of data and verify that data columns are populated with the types of information you expect. If a profile reveals problems in data, you can define steps in your project to fix those problems. For example, if a profile reveals that a column contains values of greater than expected length, you can design data quality processes to remove or fix the problem values.

A profile that analyzes the data quality of selected columns is called a column profile.

Note: You can also use the Developer tool to discover primary key, foreign key, and functional dependency relationships, and to analyze join conditions on data columns.

A column profile provides the following facts about data:

- The number of distinct and null values in each column, expressed as a number and a percentage.
- The patterns of data in each column, and the frequencies with which these values occur.
- Statistics about the column values, such as the maximum and minimum lengths of values and the first and last values in each column.
- For join analysis profiles, the degree of overlap between two data columns, displayed as a Venn diagram and as a percentage value. Use join analysis profiles to identify possible problems with column join conditions.

You can run a column profile at any stage in a project to measure data quality and to verify that changes to the data meet your project objectives. You can run a column profile on a transformation in a mapping to indicate the effect that the transformation will have on data.

Story

HypoStores wants to verify that customer data is free from errors, inconsistencies, and duplicate information. Before HypoStores designs the processes to deliver the data quality objectives, it needs to measure the quality of its source data files and confirm that the data is ready to process.

Objectives

In this lesson, you complete the following tasks:

- Perform a join analysis on the `Boston_Customers` data source and the `LA_Customers` data source.
- View the results of the join analysis to determine whether or not you can successfully merge data from the two offices.
- Run a column profile on the `All_Customers` data source.
- View the column profiling results to observe the values and patterns contained in the data.

Prerequisites

Before you start this lesson, verify the following prerequisite:

- You have completed lessons 1 and 2 in this tutorial.

Time Required

- Set aside 20 minutes to complete this lesson.

Task 1. Perform a Join Analysis on Two Data Sources

In this task, you perform a join analysis on the Boston_Customers and LA_Customers data sources to view the join conditions.

1. Select the tutorial folder and click **File** > **New** > **Profile**.
2. Select **Enterprise Discovery Profile**.
3. Click **Next**.
4. In the **Name** field, enter **Tutorial_Profile**.
5. Click **Finish**.

    The Tutorial_Profile profile appears in the Object Explorer.
6. Drag the **Boston_Customers** and **LA_Customers** data sources to the editor on the right.

    Tip: Hold down the Shift key to select multiple data objects.
7. Right-click a data object name and select **Join Profile**.

    The **New Join Profile** wizard appears.
8. In the **Name** field, enter **JoinAnalysis**.
9. Verify that Boston_Customers and LA_Customers appear as data objects, and click **Next**.
10. Verify that the **CustomerID** column is selected in both data sources.

Scroll down the wizard pane to view the columns in both data sets.

Click **Next**.

11. Click **Add** to add join conditions.

The **Join Condition** dialog box appears.

12. In the **Columns** section, click **Add row**.

13. Double-click the first row in the left column and select **CustomerID**.

14. Double-click the first row in the right column and select **CustomerID**.

15. Click **OK**, and click **Finish**.

16. If the Developer tools prompts you to save the changes, click **Yes**.

The Developer tool runs the profile.

Note: Do not close the profile. You view the profile results in the next task.

Task 2. View Join Analysis Results

In this task, you view the join analysis results in the Join Result view of the JoinAnalysis profile.

1. Click the **JoinAnalysis** tab in the editor.

2. In the **Join Result** section, click the first row.

The **Details** section displays a Venn diagram and the color key that details the results of the join analysis.

3. Verify that the **Join Rows** column shows zero as the number of rows that contain a join.
This value indicates that CustomerID fields do not have duplicates. You can successfully merge the two data sources.

4. To view the CustomerID values for the LA_Customers data object, double-click the circle named **LA_Customers** in the Venn diagram.

Tip: Double-click the circles in the Venn diagram to view the data rows. If the circles intersect in the Venn diagram, double-click the intersection to view data values common to both data sets.

The Data Viewer displays the CustomerID values from the LA_Customers data object.

Task 3. Run a Profile on a Data Source

In this task, you run a profile on the All_Customers data source to view the content and structure of the data.

1. In the **Object Explorer** view, browse to the data objects in your tutorial project.

2. Select the **All_Customers** data source.

3. Click **File** > **New** > **Profile**.

The **New** dialog box appears.

4. Select **Profile**.

5. Click **Next**.

6. In the **Name** field, enter **All_Customers**.

7. Click **Finish**.

   The All_Customers profile opens in the editor and the profile runs.

Task 4. View Column Profiling Results

In this task, you view the column profiling results for the All_Customers data object and examine the values and patterns contained in the data.

1. Click **Window** > **Show View** > **Progress** to view the progress of the All_Customers profile.

   The Progress view opens.

2. When the Progress view reports that the All_Customers profile finishes running, click the **Results** view in the editor.

3. In the **Column Profiling** section, click the **CustomerTier** column.

   The **Details** section displays all values contained in the CustomerTier column and displays information about how frequently the values occur in the data set.

4. In the **Details** section, double-click **Ruby**.

   The Data Viewer runs and displays the records where the CustomerTier column contains the value **Ruby**.

5. In the **Column Profiling** section, click the **OrderAmount** column.

6. In the **Details** section, click the **Show** list and select **Patterns**.

   The **Details** section shows the patterns found in the OrderAmount column. The string **9(5)** in the Pattern column refers to records that contain five-figure order amounts. The string **9(4)** refers to records containing four-figure order amounts.

7. In the **Pattern** column, double-click the string **9(4)**.

   The Data Viewer runs and displays the records where the OrderAmount column contains a four-figure order amount.

8. In the **Details** section, click the **Show** list and select **Statistics**.

   The **Details** section shows statistics for the OrderAmount column including the average value, standard deviation, maximum and minimum lengths, the five most common values, and the five least common values.

Profiling Data Summary

In this lesson, you learned that a profile provides information about the content and structure of the data.

You learned that you can perform a join analysis on two data objects and view the degree of overlap between the data objects. You also learned that you can run a column profile on a data object and view values, patterns, and statistics that relate to each column in the data object.

You created the `JoinAnalysis` profile to determine whether data from the `Boston_Customers` data object can merge with the data in the `LA_Customers` data object. You viewed the results of this profile and determined that all values in the `CustomerID` column are unique and that you can merge the data objects successfully.

You created the `All_Customers` profile and ran a column profile on the `All_Customers` data object. You viewed the results of this profile to discover values, patterns, and statistics for columns in the `All_Customers`

data object. Finally, you ran the Data Viewer to view rows containing values and patterns that you selected, enabling you to verify the quality of the data.

Lesson 4. Parsing Data

This chapter includes the following topics:

Parsing Data Overview

You parse data to identify one or more data elements in an input field and to write each element to a different output field.

Parsing allows you to have greater control over the information in each column. For example, consider a data field that contains a person's full name, `Bob Smith`. You can use the Parser transformation to split the full name into separate data columns for the first name and last name. After you parse the data into new columns, you can create custom data quality operations for each column.

You can configure the Parser transformation to use *token sets* to parse data columns into component strings. A token set identifies data elements such as words, ZIP codes, phone numbers, and Social Security numbers.

You can also use the Parser transformation to parse data that matches reference table entries or custom regular expressions that you enter.

Story

HypoStores wants the format of customer data files from the Los Angeles office to match the format of the data files from the Boston office. The customer data from the Los Angeles office stores the customer name in a `FullName` column, while the customer data from the Boston office stores the customer name in separate `FirstName` and `LastName` columns. HypoStores needs to parse the Los Angeles `FullName` column data into first names and last names so that the format of the Los Angeles data will match the format of the Boston data.

Objectives

In this lesson, you complete the following tasks:

- Create and configure an `LA_Customers_tgt` data object to contain parsed data.
- Create a mapping to parse the `FullName` column into separate `FirstName` and `LastName` columns.
- Add the `LA_Customers` data object to the mapping to connect to the source data.
- Add the `LA_Customers_tgt` data object to the mapping to create a target data object.
- Add a Parser transformation to the mapping and configure it to use a token set to parse full names into first names and last names.
- Run a profile on the Parser transformation to review the data before you generate the target data source.
- Run the mapping to generate parsed names.
- Run the Data Viewer to view the mapping output.

Prerequisites

Before you start this lesson, verify the following prerequisite:

- You have completed lessons 1 and 2 in this tutorial.

Timing

Set aside 20 minutes to complete the tasks in this lesson.

Task 1. Create a Target Data Object

In this task, you create an `LA_Customers_tgt` data object that you can write parsed names to.

To create a target data object, complete the following steps:

1. Create an `LA_Customers_tgt` data object based on the `LA_Customers.csv` file.
2. Configure the read and write options for the data object, including file locations and file names.
3. Add `Firstname` and `Lastname` columns to the `LA_Customers_tgt` data object.

Step 1. Create an LA_Customers_tgt Data Object

In this step, you create an `LA_Customers_tgt` data object based on the `LA_Customers.csv` file.

1. Click **File** > **New** > **Data Object**.

   The **New** window opens.
2. Select **Flat File Data Object** and click **Next**.
3. Verify that **Create from an existing flat file** is selected.
4. Click **Browse** and navigate to `LA_Customers.csv` in the following directory: `<Informatica Installation Directory>\clients\DeveloperClient\Tutorials`
5. Click **Open**.
6. In the `Name` field, enter `LA_Customers_tgt`.
7. Click **Next**.
8. Click **Next**.

9. In the **Preview Options** section, select **Import column names from first line** and click **Next**.

10. Click **Finish**.

   The `LA_Customers_tgt` data object appears in the editor.

Step 2. Configure Read and Write Options

In this step, you configure the read and write options for the `LA_Customers_tgt` data object, including file locations and file names.

1. Verify that the `LA_Customers_tgt` data object is open in the editor.

2. In the editor, select the **Read** view.

3. Click **Window** > **Show View** > **Properties**.

4. In the **Properties** view, select the **Runtime** view.

5. In the **Value** column, double-click the source file name and type `LA_Customers_tgt.csv`.

6. In the **Value** column, double-click to highlight the source file directory.

7. Right-click the highlighted name and select **Copy.**

8. In the editor, select the **Write** view.

9. In the **Properties** view, select the **Runtime** view.

10. In the **Value** column, double-click the **Output file directory** entry.

11. Right-click and select **Paste** to paste the directory location you copied from the **Read** view.

12. In the **Value** column, double-click the **Header options** entry and choose `Output Field Names`.

13. In the **Value** column, double-click the **Output file name** entry and type `LA_Customers_tgt.csv`.

14. Click **File** > **Save** to save the data object.

Step 3. Add Columns to the Data Object

In this step, you add `Firstname` and `Lastname` columns to the `LA_Customers_tgt` data object.

1. In the Object Explorer view, browse to the data objects in your tutorial project.

2. Double-click the `LA_Customers_tgt` data object.

   The `LA_Customers_tgt` data object opens in the editor.

3. Verify that the **Overview** view is selected.

4. Select the **FullName** column and click the New button to add a column.

   A column named `FullName1` appears.

5. Rename the column to `Firstname`. Click the **Precision** field and enter "30."

6. Select the `Firstname` column and click the New button to add a column.

   A column named `FirstName1` appears.

7. Rename the column to `Lastname`. Click the **Precision** field and enter "30."

8. Click **File** > **Save** to save the data object .

Task 2. Create a Mapping to Parse Data

In this task, you create a mapping and configure it to use data objects and a Parser transformation.

To create a mapping to parse data, complete the following steps:

1. Create a mapping.
2. Add source and target data objects to the mapping.
3. Add a Parser transformation to the mapping.
4. Configure the Parser transformation to parse the source column containing the full customer name into separate target columns containing the first name and last name.

Step 1. Create a Mapping

In this step, you create and name the mapping.

1. In the Object Explorer view, select your tutorial project.
2. Click **File** > **New** > **Mapping**.

   The **New Mapping** window opens.
3. In the **Name** field, enter `ParserMapping`.
4. Click **Finish**.

   The mapping opens in the editor.

Step 2. Add Data Objects to the Mapping

In this step, you add the `LA_Customers` data object and the `LA_Customers_tgt` data object to the mapping.

1. In the Object Explorer view, browse to the data objects in your tutorial project.
2. Select the `LA_Customers` data object and drag it to the editor.

   The **Add Physical Data Object to Mapping** window opens.
3. Verify that **Read** is selected and click **OK**.

   The data object appears in the editor.
4. In the Object Explorer view, browse to the data objects in your tutorial project.
5. Select the `LA_Customers_tgt` data object and drag it to the editor.

   The **Add Physical Data Object to Mapping** window opens.
6. Select **Write** and click **OK**.

   The data object appears in the editor.
7. Select the `CustomerID`, `CustomerTier`, and `FullName` ports in the `LA_Customers` data object. Drag the ports to the `CustomerID` port in the `LA_Customers_tgt` data object.

   Tip: Hold down the **CTRL** key to select multiple ports.

   The ports of the `LA_Customers` data object connect to corresponding ports in the `LA_Customers_tgt` data object.

Step 3. Add a Parser Transformation to the Mapping

In this step, you add a Parser transformation to the `ParserMapping` mapping.

1. Select the editor containing the `ParserMapping` mapping.
2. In the Transformation palette, select the Parser transformation.
3. Click the editor.

    The **New Parser Transformation** window opens.
4. Verify that **Token Parser** is selected and click **Finish**.

    The Parser transformation appears in the editor.
5. Select the `FullName` port in the `LA_Customers` data object and drag the port to the `Input` group of the Parser transformation.

    The `FullName` port appears in the Parser transformation and is connected to the `FullName` port in the data object.

Step 4. Configure the Parser Transformation

In this step, you configure the Parser transformation to parse the column containing the full customer name into separate columns that contain the first name and last name.

1. Select the editor containing the `ParserMapping` mapping.
2. Click the Parser transformation.
3. Click **Window** > **Show View** > **Properties**.
4. In the Properties view, select the **Strategies** view.
5. Click **New**. The New Strategy wizard displays.
6. Click the selection arrow in the Inputs column, and choose the `FullName` port.
7. Select the character space delimiter [\s].
8. Click **Next**.
9. Select the **Parse using Token Set** operation, and click **Next**.
10. Select **Fixed Token Sets (Single Output Only)** and choose the **Undefined** token set.
11. Click the **Outputs** field and select **New**.
12. In the **Operation Outputs** dialog box, change the output name to `Undefined_Output`.
13. Click **Finish**.
14. In the Parser transformation, click the `Undefined_Output` port and drag it to the `FirstName` port in the `LA_customers_tgt` data object.

    A connection appears between the ports.
15. In the Parser transformation, click the `OverflowField` port and drag it to the `LastName` port in the `LA_customers_tgt` data object.

    A connection appears between the ports.
16. Click **File** > **Save** to save the mapping.

Task 3. Run a Profile on the Parser Transformation

In this task, you run a profile on the Parser transformation to verify that you configured the Parser transformation to parse the full name correctly.

1. Select the editor containing the `ParserMapping` mapping.

2. Right-click the Parser transformation and select **Profile Now**.

   The profile runs and opens in the editor.

3. In the editor, click the **Results** view to display the result of the profiling operation.

4. Select the `Undefined_output` column to display information about the column in the **Details** section.

   The values contained in the `Undefined_output` column appear in the **Details** section, along with frequency and percentage statistics for each value.

5. View the data and verify that only first names appear in the `Undefined_output` column.

Task 4. Run the Mapping

In this task, you run the mapping to create the mapping output.

1. Select the editor containing the `ParserMapping` mapping.

2. Click **Run** > **Run Mapping**.

   The mapping runs and writes output to the `LA_Customers_tgt.csv` file.

Task 5. View the Mapping Output

In this task, you run the Data Viewer to view the mapping output.

1. In the Object Explorer view, locate the `LA_Customers_tgt` data object in your tutorial project and double click the data object.

   The data object opens in the editor.

2. Click **Window** > **Show View** > **Data Viewer**.

   The Data Viewer view opens.

3. In the Data Viewer view, click **Run**.

   The Data Viewer runs and displays the data.

4. Verify that the `FirstName` and `LastName` columns display correctly parsed data.

Parsing Data Summary

In this lesson, you learned that parsing data identifies the data elements in an input field and writes each element to a new column.

You learned that you use the Parser transformation to parse data. You also learned that you can create a profile for a transformation in a mapping to analyze the output from that transformation. Finally, you learned that you can view mapping output using the Data Viewer.

You created and configured the `LA_Customers_tgt` data object to contain parsed output. You created a mapping to parse the data. In this mapping, you configured a Parser transformation with a token set to parse first names and last names from the `FullName` column in the Los Angeles customer file. You configured the mapping to write the parsed data to the `Firstname` and `Lastname` columns in the `LA_Customers_tgt` data object. You also ran a profile to view the output of the transformation before you ran the mapping. Finally, you ran the mapping and used the Data Viewer to view the new data columns in the `LA_Customers_tgt` data object.

CHAPTER 14

Lesson 5. Standardizing Data

This chapter includes the following topics:

Standardizing Data Overview

Standardizing data improves data quality by removing errors and inconsistencies in the data.

To improve data quality, standardize data that contains the following types of values:

- Incorrect values
- Values with correct information in the wrong format
- Values from which you want to derive new information

Use the Standardizer transformation to search for these values in data. You can choose one of the following search operation types:

- **Text.** Search for custom strings that you enter. Remove these strings or replace them with custom text.
- **Reference table.** Search for strings contained in a reference table that you select. Remove these strings, or replace them with reference table entries or custom text.

For example, you can configure the Standardizer transformation to standardize address data containing the custom strings `Street` and `St.` using the replacement string `ST.` The Standardizer transformation replaces the search terms with the term `ST.` and writes the result to a new data column.

Story

HypoStores needs to standardize its customer address data so that all addresses use terms consistently. The address data in the `All_Customers` data object contains inconsistently formatted entries for common terms such as `Street`, `Boulevard`, `Avenue`, `Drive`, and `Park`.

Objectives

In this lesson, you complete the following tasks:

- Create and configure an `All_Customers_Stdz_tgt` data object to contain standardized data.
- Create a mapping to standardize the address terms `Street`, `Boulevard`, `Avenue`, `Drive`, and `Park` to a consistent format.
- Add the `All_Customers` data object to the mapping to connect to the source data.
- Add the `All_Customers_Stdz_tgt` data object to the mapping to create a target data object.
- Add a Standardizer transformation to the mapping and configure it to standardize the address terms.
- Run the mapping to generate standardized address data.
- Run the Data Viewer to view the mapping output.

Prerequisites

Before you start this lesson, verify the following prerequisite:

- You have completed lessons 1 and 2 in this tutorial.

Timing

Set aside 15 minutes to complete this lesson.

Task 1. Create a Target Data Object

In this task, you create an `All_Customers_Stdz_tgt` data object that you can write standardized data to.

To create a target data object, complete the following steps:

1. Create an `All_Customers_Stdz_tgt` data object based on the `All_Customers.csv` file.
2. Configure the read and write options for the data object, including file locations and file names.

Step 1. Create an All_Customers_Stdz_tgt Data Object

In this step, you create an `All_Customers_Stdz_tgt` data object based on the All_Customers.csv file.

1. Click **File** > **New** > **Data Object**.

   The **New** window opens.
2. Select **Flat File Data Object** and click **Next**.
3. Verify that **Create from an existing flat file** is selected.
4. Click **Browse** and navigate to `All_Customers.csv` in the following directory: `<Informatica Installation Directory>\clients\DeveloperClient\Tutorials`
5. Click **Open**.
6. In the `Name` field, enter `All_Customers_Stdz_tgt`.
7. Click **Next**.
8. Click **Next**.
9. In the **Preview Options** section, select **Import column names from first line** and click **Next**.

10. Click **Finish**.

   The `All_Customers_Stdz_tgt` data object appears in the editor.

Step 2. Configure Read and Write Options

In this step, you configure the read and write options for the `All_Customers_Stdz_tgt` data object, including file locations and file names.

1. Verify that the `All_Customers_Stdz_tgt` data object is open in the editor.

2. In the editor, select the **Read** view.

3. Click **Window** > **Show View** > **Properties**.

4. In the **Properties** view, select the **Runtime** view.

5. In the **Value** column, double-click the source file name and type `All_Customers_Stdz_tgt.csv`.

6. In the **Value** column, double-click the **Source file directory** entry.

7. Right-click the highlighted name and select **Copy**.

8. In the editor, select the **Write** view.

9. In the **Properties** view, select the **Runtime** view.

10. In the **Value** column, double-click the **Output file directory** entry.

11. Right-click and select **Paste** to paste the directory location you copied from the **Read** view.

12. In the **Value** column, double-click the **Header options** entry and choose `Output Field Names`.

13. In the **Value** column, double-click the **Output file name** entry and type `All_Customers_Stdz_tgt.csv`.

14. Click **File** > **Save** to save the data object.

Task 2. Create a Mapping to Standardize Data

In this task, you create a mapping and configure the mapping to use data objects and a Standardizer transformation.

To create a mapping to standardize data, complete the following steps:

1. Create a mapping.

2. Add source and target data objects to the mapping.

3. Add a Standardizer transformation to the mapping.

4. Configure the Standardizer transformation to standardize common address terms to consistent formats.

Step 1. Create a Mapping

In this step, you create and name the mapping.

1. In the **Object Explorer** view, select your tutorial project.

2. Click **File** > **New** > **Mapping**.

   The **New Mapping** window opens.

3. In the **Name** field, enter `StandardizerMapping`.

4. Click **Finish**.

    The mapping opens in the editor.

Step 2. Add Data Objects to the Mapping

In this step, you add the `All_Customers` data object and the `All_Customers_Stdz_tgt` data object to the mapping.

1. In the **Object Explorer** view, browse to the data objects in your tutorial project.

2. Select the `All_Customers` data object and drag it to the editor.

    The **Add Physical Data Object to Mapping** window opens.

3. Verify that **Read** is selected and click **OK**.

    The data object appears in the editor.

4. In the **Object Explorer** view, browse to the data objects in your tutorial project.

5. Select the `All_Customers_Stdz_tgt` data object and drag it to the editor.

    The **Add Physical Data Object to Mapping** window opens.

6. Select **Write** and click **OK**.

    The data object appears in the editor.

7. Select all ports in the `All_Customers` data object. Drag the ports to the `CustomerID` port in the `All_Customers_Stdz_tgt` data object.

    Tip: Hold down the Shift key to select multiple ports. You might need to scroll down the list of ports to select all of them.

    The ports of the `All_Customers` data object connect to corresponding ports in the `All_Customers_Stdz_tgt` data object.

Step 3. Add a Standardizer Transformation to the Mapping

In this step, you add a Standardizer transformation to standardize strings in the address data.

1. Select the editor that contains the `StandardizerMapping` mapping.

2. In the Transformation palette, select the Standardizer transformation.

3. Click the editor.

    A Standardizer transformation named `NewStandardizer` appears in the mapping.

4. To rename the Standardizer transformation, double-click the title bar of the transformation and type `AddressStandardizer`.

5. Select the `Address1` port in the `All_Customers` data object, and drag the port to the **Input** group of the Standardizer transformation.

    A port named `Address1` appears in the input group. The port connects to the `Address1` port in the `All_Customers` data object.

Note: You add an output port to the transformation when you configure a standardization strategy.

Step 4. Configure the Standardizer Transformation

In this step, you configure the Standardizer transformation to standardize address terms in the source data.

Note: You will define five standardization operations in this task. Each operation replaces a string in the input column with a new string.

1. Select the editor that contains the `StandardizerMapping` mapping.
2. Click the Standardizer transformation.
3. Click **Window** > **Show View** > **Properties**.
4. In the Properties view, select **Strategies**.
5. Click **New**. The New Strategy wizard displays.
6. Click the selection arrow in the Inputs column, and choose the `Address1` input port.

    The Outputs field shows `Address1` as the output port.
7. Select the character space and comma delimiters [\s] and [,]. Optionally, select the options to remove trailing spaces.
8. Click **Next**.
9. Select the **Replace custom strings** operation, and click **Next**.
10. Under **Properties**, click **New**.
11. Edit the **Custom Strings** and **Replace With** fields so that they contain the first pair of strings from the following table:

Custom Strings	Replace With
STREET	ST.
BOULEVARD	BLVD.
AVENUE	AVE.
DRIVE	DR.
PARK	PK.

12. Repeat steps 9 through 12 to define standardization operations for all strings in the table.
13. Drag the `Address1` output port to the `Address1` port in the All_Customers_Stdz_tgt data object.
14. Click **File** > **Save** to save the mapping.

Task 3. Run the Mapping

In this task, you run the mapping to write standardized addresses to the output data object.

1. Select the editor containing the `StandardizerMapping` mapping.
2. Click **Run** > **Run Mapping**.

    The mapping runs and writes output to the `All_Customers_Stdz_tgt.csv` file.

Task 4. View the Mapping Output

In this task, you run the data viewer to view the mapping output and verify that the address data is correctly standardized.

1. In the **Object Explorer** view, locate the `All_Customers_Stdz_tgt` data object in your tutorial project and double-click the data object.

   The data object opens in the editor.

2. Click **Window** > **Show View** > **Data Viewer**.

   The Data Viewer view opens.

3. In the Data Viewer view, click **Run**.

   The Data Viewer displays the mapping output.

4. Verify that the `Address1` column displays correctly standardized data. For example, all instances of the string `STREET` should be replaced with the string `ST`.

Standardizing Data Summary

In this lesson, you learned that you can standardize data to remove errors and inconsistencies in the data.

You learned that you can use a Standardizer transformation to standardize strings in an input column. You also learned that you can view mapping output using the Data Viewer.

You created and configured the `All_Customers_Stdz_tgt` data object to contain standardized output. You created a mapping to standardize the data. In this mapping, you configured a Standardizer transformation to standardize the `Address1` column in the `All_Customers` data object. You configured the mapping to write the standardized output to the `All_Customers_Stdz_tgt` data object. Finally, you ran the mapping and used the Data Viewer to view the standardized data in the `All_Customers_Stdz_tgt` data object.

Lesson 6. Validating Address Data

This chapter includes the following topics:

Validating Address Data Overview

Address validation is the process of evaluating and improving the quality of postal addresses. It evaluates address quality by comparing input addresses with a reference dataset of valid addresses. It improves address quality by identifying incorrect address values and using the reference dataset to create fields that contain correct values.

An address is valid when it is deliverable. An address may be well formatted and contain real street, city, and post code information, but if the data does not result in a deliverable address then the address is not valid. The Developer tool uses address reference datasets to check the deliverability of input addresses. Informatica provides address reference datasets.

An address reference dataset contains data that describes all deliverable addresses in a country. The address validation process searches the reference dataset for the address that most closely resembles the input address data. If the process finds a close match in the reference dataset, it writes new values for any incorrect or incomplete data values. The process creates a set of alphanumeric codes that describe the type of match found between the input address and the reference addresses. It can also restructure the address, and it can add information that is absent from the input address, such as a four-digit ZIP code suffix for a United States address.

Use the Address Validator transformation to build address validation processes in the Developer tool. This multi-group transformation contains a set of predefined input ports and output ports that correspond to all possible fields in an input address. When you configure an Address Validator transformation, you select the default reference dataset, and you create an input and output address structure using the transformation ports. In this lesson you configure the transformation to validate United States address data.

Story

HypoStores needs correct and complete address data to ensure that its direct mail campaigns and other consumer mail items reach its customers. Correct and complete address data also reduces the cost of mailing operations for the organization. In addition, HypoStores needs its customer data to include addresses in a printable format that is flexible enough to include addresses of different lengths.

To meet these business requirements, the HypoStores ICC team creates an address validation mapping in the Developer tool.

Objectives

In this lesson, you complete the following tasks:

- Create a target data object that will contain the validated address fields and match codes.
- Create a mapping with a source data object, a target data object, and an Address Validator transformation.
- Configure the Address Validator transformation to validate the address data of your customers.
- Run the mapping to validate the address data, and review the match code outputs to verify the validity of the address data.

Prerequisites

Before you start this lesson, verify the following prerequisites:

- You have completed lessons 1 and 2 in this tutorial.
- United States address reference data is installed in the domain and registered with the Administrator tool. Contact your Informatica administrator to verify that United States address data is installed on your system. The reference data installs through the Data Quality Content Installer.

Timing

Set aside 25 minutes to complete this lesson.

Task 1. Create a Target Data Object

In this task, you create a target data object, configure the write options, and add ports.

To create and configure the target data object, complete the following steps:

1. Create an All_Customers_av_tgt data object based on the All_Customers.csv file.
2. Configure the read and write options for the data object, including the file locations and file names.
3. Add ports to the data object to receive the match code values generated by the Address Validator transformation.

Step 1. Create the All_Customers_av_tgt Data Object

In this step, you create an `All_Customers_av_tgt` data object based on the `All_Customers.csv` file.

1. Click **File** > **New** > **Data Object**.

   The **New** window opens.
2. Select **Flat File Data Object** and click **Next**.

3. Verify that **Create from an existing flat file** is selected. Click **Browse** next to this selection, find the `All_Customers.csv` file, and click **Open**.

4. In the `Name` field, enter `All_Customers_av_tgt`.

5. Click **Next**.

6. Click **Next**.

7. In the **Preview Options** section, select **Import column names from first line** and click **Next**.

8. Click **Finish**.

   The `All_Customers_av_tgt` data object appears in the editor.

Step 2. Configure Read and Write Options

In this step, you configure the read and write options for the `All_Customers_av_tgt` data object, including the target file name and location.

1. Verify that the `All_Customers_av_tgt` data object is open in the editor.

2. In the editor, select the **Read** view.

3. Select **Window** > **Show View** > **Properties**.

4. In the **Properties** view, select the **Runtime** view.

5. In the **Value** column, double-click the source file name and type `All_Customers_av_tgt.csv`.

6. In the **Value** column, double-click to highlight the source file directory path.

7. Right-click the highlighted path and name and select **Copy.**

8. In the editor, select the **Write** view.

9. In the **Properties** view, select the **Runtime** view.

10. In the **Value** column, double-click the **Output file directory** entry.

11. Right-click this entry and select **Paste** to add the path you copied from the **Read** view.

12. In the **Value** column, double-click the **Header options** entry and choose `Output Field Names`.

13. In the **Value** column, double-click the **Output file name** entry and type `All_Customers_av_tgt.csv`.

14. Select **File** > **Save** to save the data object.

Step 3. Add Ports to the Data Object

In this step, you add two ports to the `All_Customers_av_tgt` data object so that the Address Validator transformation can write match code values to the target file. Name the ports `MailabilityScore` and `MatchCode`.

The `MailabilityScore` value describes the deliverability of the input address. The `MatchCode` value describes the type of match the transformation makes between the input address and the reference data addresses.

1. In the **Object Explorer** view, browse to the data objects in your tutorial project.

2. Double-click the `All_Customers_av_tgt` data object.

   The `All_Customers_av_tgt` data object opens in the editor.

3. Verify that **Overview** is selected.

4. Select the final port in the port list. This port is named `MiscDate`.

5. Click **New**.

   A port named `MiscDate1` appears.

6. Rename the `MiscDate1` port to `MailabilityScore`.

7. Select the `MailabilityScore` port.

8. Click **New**.

   A port named `MailabilityScore1` appears.

9. Rename the `MailabilityScore1`port to `MatchCode`.

10. Click **File** > **Save** to save the data object.

Task 2. Create a Mapping to Validate Addresses

In this task, you create a mapping and add data objects and an Address Validator transformation.

To create the mapping and add the objects you need, complete the following steps:

1. Create a mapping object.

2. Add source and target data objects to the mapping.

3. Add an Address Validator transformation to the mapping.

Step 1. Create a Mapping

In this step, you create and name the mapping.

1. In the Object Explorer view, select your tutorial project.

2. Select **File** > **New** > **Mapping**.

   The **New Mapping** window opens.

3. In the **Name** field, enter `ValidateAddresses`.

4. Click **Finish**.

   The mapping opens in the editor.

Step 2. Add Data Objects to the Mapping

In this step, you add the source and target data objects to the mapping.

`All_Customers` is the source data object for the mapping. The Address Validator transformation reads data from this object. `All_Customers_av_tgt` is the data target object for the mapping. This object reads data from the Address Validator transformation.

1. In the **Object Explorer** view, browse to the data objects in your tutorial project.

2. Select the `All_Customers` data object and drag it to the editor.

   The **Add Physical Data Object to Mapping** window opens.

3. Verify that **Read** is selected and click **OK**.

   The data object appears in the editor.

4. In the **Object Explorer** view, browse to the data objects in your tutorial project.

5. Select the `All_Customers_av_tgt` data object and drag it onto the editor.

   The **Add Physical Data Object to Mapping** window opens.

6. Select **Write** and click **OK**.

    The data object appears in the editor.

7. Click **Save**.

Step 3. Add an Address Validator Transformation to the Mapping

In this step, you add an Address Validator transformation to the mapping that contains the source and data objects.

When this step is complete, you can configure the transformation and connect its ports to the data objects.

1. Select the editor containing the `ValidateAddresses` mapping.

2. In the Transformation palette, select the Address Validator transformation.

3. Click the editor.

    The Address Validator transformation appears in the editor.

Task 3. Configure the Address Validator Transformation

In this task, you configure the Address Validator transformation to read and validate addresses from the `All_Customers` data source.

Note: The Address Validator transformation contain a series of predefined input and output ports. Select the ports you need and connect them to the objects in the mapping.

To configure the transformation, complete the following steps:

1. Select the default country for address validation.

2. Configure the transformation input ports.

3. Configure the transformation output ports.

4. Connect unused source ports to the data target.

Step 1. Set the Default Country for Address Validation

In this step, you select the default country for address validation. The address reference data files that the Address Validator transformation uses are organized by country. When you select the default country, you identify the country dataset that the transformation applies to any input address that does not contain country information.

1. Select the Address Validator transformation in the editor.

2. Under **Properties**, click **General Settings**.

3. In the **Default Country** menu, select `United States`.

Step 2. Configure the Address Validator Transformation Input Ports

In this step, you select transformation input ports and connect these ports to the `All_Customers_av` data object.

The Address Validator transformation contains several groups of predefined input ports. Select the input ports that correspond to the fields in your input address and add these ports to the transformation.

Hold the Ctrl key when selecting ports in the steps below to select multiple ports in a single operation.

1. Select the Address Validator transformation in the editor.
2. Under **Properties**, click **Templates**.
3. Expand the **Basic Model** port group.
4. Expand the **Hybrid** input port group and select the following ports:

Port Name	Description
Delivery Address Line 1	Street address data, such as street name and building number.
Locality Complete 1	City or town name.
Postcode 1	Postcode or ZIP code.
Province 1	Province or state name.
Country Name	Country name or abbreviation.

Note: Hold the Ctrl key to select multiple ports in a single operation.

5. On the toolbar above the port names list, click **Add port to transformation**.

   This toolbar is visible when you select **Templates**.

   The selected ports appear in the transformation in the mapping editor.
6. Connect the source ports to the Address Validator transformation ports as follows:

Source Port	Address Validator Transformation Port
Address1	Delivery Address Line 1
City	Locality Complete 1
ZIP	Postcode 1
State	Province 1
Country	Country Name

Step 3. Configure the Address Validator Transformation Output Ports

In this step, you select transformation output ports and connect these ports to the `All_Customers_av_tgt` data object.

The Address Validator transformation contains several groups of predefined output ports. Select the ports that define the address structure you require and add these ports to the transformation.

You can also select ports containing information on the type of validation achieved for each address.

1. Select the Address Validator transformation in the mapping editor.
2. Under **Properties**, click **Templates**.
3. Expand the **Basic Model** port group.
4. Expand the **Address Elements** output port group and select the following port:

Port Name	Description
Street Complete 1	Street address data, such as street name and building number.

5. Expand the **Last Line Elements** output port group and select the following ports:

Port Name	Description
Locality Complete 1	City or town name.
Postcode 1	Postcode or ZIP code.
Province Abbreviation 1	Province or state identifier.

Note: Hold the Ctrl key to select multiple ports in a single operation.

6. Expand the **Country** output port group and select the following port:

Port Name	Description
Country Name 1	Country name.

7. Expand the **Status Info** output port group and select the following ports:

Port Name	Description
Mailability Score	Score that represents the chance of successful postal delivery.
Match Code	Code that represents the degree of similarity between the input address and the reference data.

8. On the toolbar above the port names list, click **Add port to transformation**.

   This toolbar is visible when you select **Templates**.

9. Connect the Address Validator transformation ports to the `All_Customers_av_tgt` ports as follows:

Address Validator Transformation Port	Target Port
Street Complete 1	Address1
Locality Complete 1	City
Postcode 1	ZIP
Province Abbreviation 1	State
Country Name 1	Country
Mailability Score	MailabilityScore
Match Code	MatchCode

Step 4. Connect Unused Data Source Ports to the Data Target

In this step, you connect the unused ports on the `All_Customers` data source to the data target.

▶ Connect the unused ports on the data source to the ports with the same names on the data target.

Task 4. Run the Mapping

In this task, you run the mapping to create the mapping output.

1. Select the editor containing the `ValidateAddresses` mapping.

2. Select **Run** > **Run Mapping**.

    The mapping runs and writes output to the `All_Customers_av_tgt.csv` file.

Task 5. View the Mapping Output

In this task, you run the Data Viewer to view the mapping output. Review the quality of your validated addresses by examining the values written to the Mailability Score and Match Code columns in the target data object.

The Match Code value is an alphanumeric code representing the type of validation that the mapping performed on the address.

The Mailability Score value is a single-digit value that summarizes the deliverability of the address.

1. In the Object Explorer view, find the `All_Customers_av_tgt` data object in your tutorial project and double click the data object.

    The data object opens in the editor.

2. Select **Window** > **Show View** > **Data Viewer**.

    The Data Viewer opens.

3. In the Data Viewer, click **Run**.

    The Data Viewer displays the mapping output.

4. Scroll across the mapping results so that the Mailability Score and Match Code columns are visible.

5. Review the values in the Mailability Score column.

    The scores can range from 0 through 5. Addresses with higher scores are more likely to be delivered successfully.

6. Review the values in the Match Code column.

    Match Code is an alphanumeric code. The alphabetic character indicates the type of validation that the transformation performed, and the digit indicates the quality of the final address.

    The following table describes the Match Code values:

Code	Description
A1	Address code lookup found a partial address or a complete address for the input code.
A0	Address code lookup found no address for the input code.
C4	Corrected. All postally relevant elements are checked.
C3	Corrected. Some elements cannot be checked.
C2	Corrected, but the delivery status is unclear due to absent reference data.
C1	Corrected, but the delivery status is unclear because user standardization introduced errors.
I4	Data cannot be corrected completely, but there is a single match with an address in the reference data.
I3	Data cannot be corrected completely, and there are multiple matches with addresses in the reference data.
I2	Data cannot be corrected. Batch mode returns partial suggested addresses.
I1	Data cannot be corrected. Batch mode cannot suggest an address.
N7	Validation error. Validation did not take place because single-line validation is not unlocked.
N6	Validation error. Validation did not take place because single-line validation is not supported for the destination country.
N5	Validation error. Validation did not take place because the reference database is out of date.
N4	Validation error. Validation did not take place because the reference data is corrupt or badly formatted.
N3	Validation error. Validation did not take place because the country data cannot be unlocked.

Code	Description
N2	Validation error. Validation did not take place because the required reference database is not available.
N1	Validation error. Validation did not take place because the country is not recognized or not supported.
Q3	Suggestion List mode. Address validation can retrieve one or more complete addresses from the address reference data that correspond to the input address.
Q2	Suggestion List mode. Address validation can combine the input address elements and elements from the address reference data to create a complete address.
Q1	Suggestion List mode. Address validation cannot suggest a complete address. To generate a complete address suggestion, add data to the input address.
Q0	Suggestion List mode. There is insufficient input data to generate a suggestion.
RB	Country recognized from abbreviation. Recognizes ISO two-character and ISO three-character country codes. Can also recognize common abbreviations such as "GER" for Germany.
RA	Country recognized from the Force Country setting in the transformation.
R9	Country recognized from the Default Country setting in the transformation.
R8	Country recognized from the country name.
R7	Country recognized from the country name, but the transformation identified errors in the country data.
R6	Country recognized from territory data.
R5	Country recognized from province data.
R4	Country recognized from major town data.
R3	Country recognized from the address format.
R2	Country recognized from a script.
R1	Country not recognized because multiple matches are available.
R0	Country not recognized.
S4	Parse mode. The address was parsed perfectly.
S3	Parse mode. The address was parsed with multiple results.
S1	Parse mode. There was a parsing error due to an input format mismatch.
V4	Verified. The input data is correct. Address validation checked all postally relevant elements, and inputs matched perfectly.

Code	Description
V3	Verified. The input data is correct, but some or all elements were standardized, or the input contains outdated names or exonyms.
V2	Verified. The input data is correct, but some elements cannot be verified because of incomplete reference data.
V1	Verified. The input data is correct, but user standardization has negatively impacted deliverability. For example, the post code length is too short.

Validating Address Data Summary

In this lesson, you learned that address validation compares input address data with reference data and returns the most accurate possible version of the address.

You learned that the address validation process also returns status information on the quality of each address.

You learned that Administrator tool users run the Data Quality Content Installer to install address reference data.

You also learned that the Address Validator transformation is a multi-group transformation, and that you select the input and output ports for the transformation from the port groups. The input ports you select determine the content of the address that is validated. The output ports determine the content of the final address record.

APPENDIX A

Frequently Asked Questions

This appendix includes the following topics:

Informatica Analyst Frequently Asked Questions

Review the Frequently Asked Questions to answer questions you might have about Informatica Analyst.

Can I use one user account to access the Administrator tool, the Developer tool, and the Analyst tool?

Yes. You can give a user permission to access all three tools. You do not need to create separate user accounts for each client application.

Where is my reference data stored?

You can use the Developer tool and the Analyst tool to create and share reference data objects. The Model repository stores the reference data object metadata. The reference data database stores reference table data values. Configure the reference data database on the Content Management Service.

Informatica Developer Frequently Asked Questions

Review the frequently asked questions to answer questions you might have about Informatica Developer.

What is the difference between a mapplet and a rule?

You can validate a mapplet as a rule. A rule is business logic that defines conditions applied to source data, for example when you run a profile. You can validate a mapplet as a rule when the mapplet meets the following requirements:

- It contains an Input and Output transformation.
- It does not contain active transformations.
- It does not specify cardinality between input groups.

I have a Data Engineering product license. Can I use the Developer tool to export objects to PowerCenter?

No. Engineering products do not integrate with PowerCenter.

What is the difference between a source and target in PowerCenter and a physical data object in the Developer tool?

In PowerCenter, you create a source definition to include as a mapping source. You create a target definition to include as a mapping target. In the Developer tool, you create a physical data object that you can use as a mapping source or target.

What is the difference between a mapping in the Developer tool and a mapping in PowerCenter?

A PowerCenter mapping specifies how to move data between sources and targets. A Developer tool mapping specifies how to move data between the mapping input and output.

A PowerCenter mapping must include one or more source definitions, source qualifiers, and target definitions. A PowerCenter mapping can also include shortcuts, transformations, and mapplets.

A Developer tool mapping must include mapping input and output. A Developer tool mapping can also include transformations and mapplets.

The Developer tool has the following types of mappings:

- Mapping that moves data between sources and targets. This type of mapping differs from a PowerCenter mapping only in that it cannot use shortcuts and does not use a source qualifier.

- Logical data object mapping. A mapping in a logical data object model. A logical data object mapping can contain a logical data object as the mapping input and a data object as the mapping output. Or, it can contain one or more physical data objects as the mapping input and logical data object as the mapping output.

- Virtual table mapping. A mapping in an SQL data service. It contains a data object as the mapping input and a virtual table as the mapping output.

- Virtual stored procedure mapping. Defines a set of business logic in an SQL data service. It contains an Input Parameter transformation or physical data object as the mapping input and an Output Parameter transformation or physical data object as the mapping output.

What is the difference between a mapplet in PowerCenter and a mapplet in the Developer tool?

A mapplet in PowerCenter and in the Developer tool is a reusable object that contains a set of transformations. You can reuse the transformation logic in multiple mappings.

A PowerCenter mapplet can contain source definitions or Input transformations as the mapplet input. It must contain Output transformations as the mapplet output.

A Developer tool mapplet can contain data objects or Input transformations as the mapplet input. It can contain data objects or Output transformations as the mapplet output. A mapping in the Developer tool also includes the following features:

- You can validate a mapplet as a rule. You use a rule in a profile.

- A mapplet can contain other mapplets.

INDEX