Informatica® Cloud Data Quality
April 2024

# Parse assets

Informatica Cloud Data Quality Parse assets
April 2024
April 2024

© Copyright Informatica LLC 1998, 2024

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

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

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

Portions of this software and/or documentation are subject to copyright held by third parties.

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

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

Publication Date: 2024-04-05

# Table of Contents


**Preface** . . . . . 5

Informatica Resources . . . . . 5

Informatica Documentation . . . . . 5

Informatica Intelligent Cloud Services web site . . . . . 5

Informatica Intelligent Cloud Services Communities . . . . . 5

Informatica Intelligent Cloud Services Marketplace . . . . . 6

Informatica Knowledge Base . . . . . 6

Informatica Intelligent Cloud Services Trust Center . . . . . 6

Informatica Global Customer Support . . . . . 6

**Chapter 1: Introduction to parse assets** . . . . . 7

Parsing modes . . . . . 7

Unparsed data and overflow data . . . . . 8

Rules and guidelines for overflow and unparsed data fields . . . . . 9

Parse assets and mappings . . . . . 9

Parse assets and dimensions . . . . . 10

Parse asset properties . . . . . 10

Configuring a parse asset . . . . . 12

**Chapter 2: Custom parsing operations** . . . . . 14

Parsing with dictionaries . . . . . 14

Parsing with regular expressions . . . . . 15

Parse asset structure . . . . . 16

Dictionary options on the Configuration tab . . . . . 16

Regular expression options on the Configuration tab . . . . . 18

Configuring a dictionary step . . . . . 19

Configuring a regular expression step . . . . . 20

**Chapter 3: Pattern-based parsing operations** . . . . . 21

Pattern-based parsing options on the Configuration tab . . . . . 21

Configuration summary for pattern-based parsing . . . . . 23

Installing the asset bundle for pattern-based parsing . . . . . 24

Configuring the pre-built options for pattern-based parsing . . . . . 25

Testing and updating the pattern-based parsing configuration . . . . . 25

Pattern selection and row selection options . . . . . 27

Rules and guidelines for user-defined pattern labels . . . . . 28

**Chapter 4: Test parse assets** . . . . . 31

Testing a parse asset in Custom mode . . . . . 31

Rules and guidelines for test data . . . . . 32

**Index** . . . . . **33**

# Preface

Refer to *Parse* for information on how to parse words or strings from an input field to discrete output fields based on the types of information that the words or strings contain. You can configure a parse asset to use dictionaries and regular expressions to identify the words or strings. You can also configure a parse asset to use pattern-based logic to identify person name data values. To parse your data, add the asset to a Parse transformation in a mapping in Data Integration.

# 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 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 Intelligent Cloud Services web site

You can access the Informatica Intelligent Cloud Services web site at http://www.informatica.com/cloud.

## Informatica Intelligent Cloud Services Communities

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

Access the Informatica Intelligent Cloud Services Community at:

https://network.informatica.com/community/informatica-network/products/cloud-integration

Developers can learn more and share tips at the Cloud Developer community:

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers

## Informatica Intelligent Cloud Services Marketplace

Visit the Informatica Marketplace to try and buy Data Integration Connectors, templates, and mapplets:

https://marketplace.informatica.com/

## 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 Intelligent Cloud Services Trust Center

The Informatica Intelligent Cloud Services Trust Center provides information about Informatica security policies and real-time system availability.

You can access the trust center at https://www.informatica.com/trust-center.html.

Subscribe to the Informatica Intelligent Cloud Services Trust Center to receive upgrade, maintenance, and incident notifications. The Informatica Intelligent Cloud Services Status page displays the production status of all the Informatica cloud products. All maintenance updates are posted to this page, and during an outage, it will have the most current information. To ensure you are notified of updates and outages, you can subscribe to receive updates for a single component or all Informatica Intelligent Cloud Services components. Subscribing to all components is the best way to be certain you never miss an update.

To subscribe, on the Informatica Intelligent Cloud Services Status page, click **SUBSCRIBE TO UPDATES**. You can choose to receive notifications sent as emails, SMS text messages, webhooks, RSS feeds, or any combination of the four.

## Informatica Global Customer Support

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

To find online support resources on the Informatica Network, click **Contact Support** in the Informatica Intelligent Cloud Services Help menu to go to the **Cloud Support** page. The **Cloud Support** page includes system status information and community discussions. Log in to Informatica Network and click **Need Help** to find additional resources and to contact Informatica Global Customer Support through email.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at https://www.informatica.com/services-and-training/support-services/contact-us.html.

CHAPTER 1

# Introduction to parse assets

Use a parse asset to improve the structure of your data. A parse asset defines a set of operations that can identify discrete values in an input field and write the values to appropriate output fields.

You add a parse asset to a Parse transformation in Data Integration. When you run a mapping with a Parse transformation, the transformation reads a single input field and writes the values that it identifies to new fields based on the criteria that the parse asset specifies.

You can configure a parse operation to identify values in the following ways:

**Use a dictionary to identify values**

The operation compares the values in the input field to the values in a dictionary that the parse asset specifies. When the operation finds an input value that matches the dictionary, it writes the value to a new output column.

Use a dictionary to find values based on their content.

**Use a regular expression to identify values**

The operation applies a regular expression to the input field and finds values that match the expression logic. When the operation finds an input value that matches the expression logic, it writes the value to a new output column.

Use a regular expression to find values based on their structure.

**Use a pre-built pattern to identify values**

The operation uses pre-built parsing logic to analyze the values in the input field and assign each value to a pre-defined output field. The parse asset includes built-in parsing logic for person name data. You can add patterns to the asset to enhance the parsing logic.

A parsing operation that uses a dictionary or a regular expression is called a step. You can combine steps that use dictionaries and steps that use regular expressions in a single asset. Each step parses data values to one or more output fields that you specify. The Parse transformation creates the outputs that you specify at run time and writes data to the outputs that it creates.

A parse asset can contain a single pattern-parsing operation. The Parse transformation creates the outputs that the pattern specifies and writes data to the outputs that it creates.

## Parsing modes

When you configure a parse asset, you must select the mode in which the parsing operations will run.

You can select one of the following modes:

**Custom mode**

Select custom mode to define a parsing operation that uses a dictionary or a regular expression to analyze the input data. When you work in custom mode, you configure all elements of the parsing operation.

**Pre-built mode**

Select pre-built mode to define a parsing operation that applies built-in pattern logic to the input data. When you work in pre-built mode, you do not need to configure the parsing operation. You can optionally add one or more patterns to the asset logic in order to enhance the parsing performance.

# Unparsed data and overflow data

When a parsing step cannot identify a value in an input field, the step writes the value to a field for unparsed data. When a parsing step successfully identifies a value but cannot assign the value to the designated output field, the step writes the value to a field for overflow data. A step writes a value to an overflow field when it has already parsed another value successfully to the designated output field.

When a step assigns a value to an unparsed data field, the value remains available for parsing from its original input field by subsequent steps in the asset. When a step writes a value to an overflow field, the value is no longer available to subsequent steps in the asset. In this way, a value is not finally written to the unparsed data field until all of the steps in the asset are complete.

The number of overflow fields that the asset generates at run time depends on the types of step that you configure and the asset properties that you define. For example, you might select the **Detailed Overflow** property to enable the creation of a dedicated overflow field for each step. The **Detailed Overflow** property is available in custom mode.

## Example: using the overflow data field

You might configure a parsing step to split a field of multiple values into discrete fields based on the types of information that the values represent. You can use the fields for parsed data and overflow data to capture values that contain the same type of information.

For example, the following product description contains color and product type information:

EMERALD GREEN WOODSTAIN

The product description uses two values define the color. You can define a dictionary step to parse the color names and the product type to discrete fields. The step reads a dictionary of color names and specifies a single overflow field.

At run time, the step identifies EMERALD and GREEN as colors that match values in the dictionary. The step writes EMERALD to the field for parsed data and writes GREEN to the overflow field. The step additionally writes WOODSTAIN to the unparsed data field. As a result, the step writes the terms EMERALD, GREEN, and WOODSTAIN to discrete fields.

You can optionally merge the fields that contain colors in a downstream transformation.

## Example: using an unparsed data field to create new fields

You might configure a parsing step to write a uniform or consistent set of data values to an unparsed data field.

For example, the following contact field includes a name and a prefix:

PROFESSOR INDIRA SINGH

You define a step that uses a dictionary of prefix values to parse the prefix PROFESSOR to an output field. The step parses all other information to an unparsed data field.

Because the unparsed data field contains a set of values that uniformly represent a person name, you can use the field as person name data in downstream operations.

## Rules and guidelines for overflow and unparsed data fields

The parse asset configuration determines how the associated Parse transformation creates overflow and unparsed data fields at runtime.

Consider the following rules and guidelines about the overflow and unparsed fields:

- When you configure an asset in Custom mode, the **Detailed Overflow** option determines how the Parse transformation creates overflow fields. If you clear the option, the transformation creates a single overflow field for all overflow data from the asset. If you select the option, the transformation creates an overflow field for each step in the asset.

  Find the Detailed Overflow option in the **Parse Properties** dialog box when you configure the asset in custom mode. Open the properties from the Data Quality toolbar. The option is cleared by default.
- When you configure an asset in Pre-built mode, the locale that you select determines whether the Parse transformation creates an overflow field. The transformation creates an overflow field when you set the locale to Brazil or Portugal. The transformation does not create an overflow field in other locales.
- A Parse transformation creates a single field for all unparsed data when you configure the asset in Custom mode.
- The locale that the asset specifies in Pre-built mode determines whether the Parse transformation creates an unparsed data field. The transformation does not create an unparsed data field when you set the locale to Brazil, Canada, Portugal, or the United States.

# Parse assets and mappings

The parse assets that you create are available to the users who create mappings in Data Integration. You or other users add a Parse transformation to a mapping and add a parse asset to the transformation. The Parse transformation applies the logic in the parse asset to an input data field that you specify.

A Parse transformation works in a similar way to a Mapplet transformation. A Data Integration user connects the transformation inputs and outputs to other objects in the mapping in the same manner as a mapplet connects to other mapping objects. When the Data Integration user runs the mapping, the Parse transformation applies the parse asset logic to the input field and generates output fields that you can connect to a downstream transformation.

After the mapping runs, you can evaluate the data output to determine if the data meets the parsing objectives.

### Rules and guidelines for parse assets and mappings

Consider the following rules and guidelines when you work with mappings and parse assets:

- A Parse transformation contains a single parse asset. You configure the transformation to perform parsing operations on a single field of input data. You select the field on the **Field Mappings** tab in the transformation.
- When you run a mapping that includes a Parse transformation, the mapping creates the output fields that the parse asset specifies.

When the parse asset specifies a regular expression or a dictionary, the Parse transformation creates an output field for the results of each step. The transformation also creates a field for unparsed data and one or more fields for overflow data.

When the parse asset specifies pattern-based parsing, the Parse transformation creates multiple output fields for name data, including discrete fields for first names, family names, and full names. The transformation can also create fields for overflow data and for unparsed data.

For more information about the creation of overflow and unparsed data fields, see "Unparsed data and overflow data" on page 8.

- A parse asset is a reusable object. A Data Integration user can add a single parse asset to transformations in multiple mappings.

  A Parse transformation is a nonreusable object. You select a Parse transformation when you configure a mapping.
- A parse asset is a read-only object in Data Integration. The parse asset does not change unless you update it in Data Quality.
- To view the list of parse assets that you created, select the **Explore** option in Data Quality.
- If you update a parse asset that you previously added to a transformation, you must synchronize the transformation with the parse asset.

  To synchronize the transformation, open the transformation in the mapping and select the Parse properties. Data Integration displays a message that prompts you to synchronize the transformation with the parse asset.

# Parse assets and dimensions

The data quality issues that you may find in your data can fall into a range of common categories. Data Quality assets can identify the categories as dimensions. When you configure an asset in Data Quality, you can use the **Dimension** property to indicate the type of data quality issue that you want the asset to examine.

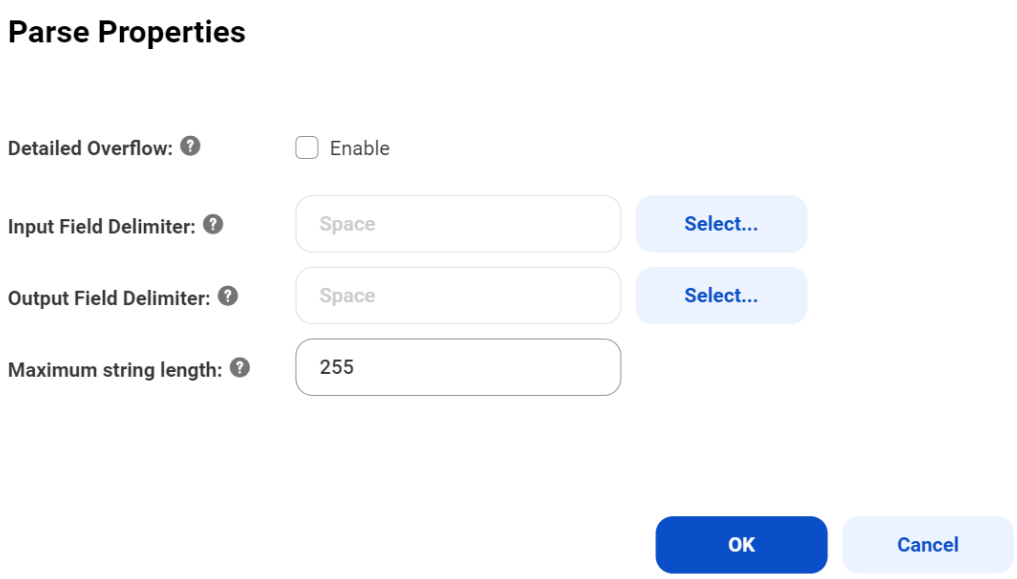
Find the Dimension property on the **Definition** tab of the asset.

For more information about data quality dimensions, see the *Introduction* in Data Quality documentation.

# Parse asset properties

You can configure properties on a parse asset that determine how a mapping that includes the asset will read, write, and process data. To view the properties, open the **Parse Properties** dialog box from the Actions menu on the Data Quality toolbar.

The following image shows the properties:




You can set the following properties:

## Detailed Overflow

Determines whether the parsing operation creates an Overflow field for each step in the asset at run time. Select or clear the option in Custom mode.

When you select the option, the transformation writes overflow data to a separate field for each step in the asset. When you clear the option, the transformation writes all overflow data that the asset identifies to a single field. The option is cleared by default.

The Detailed Overflow option does not apply in Pre-built mode, as the asset specifies a single parsing operation in Pre-built mode.

## Input Field Delimiter

Specifies the delimiter that the parsing operation recognizes when you parse data with a dictionary or regular expression. Set the option in Custom mode. Use the **Select** option to select the delimiter or to add a custom delimiter. You can select one or more delimiters. The default input delimiter is a character space.

## Output Field Delimiter

Specifies the field delimiter that the parsing operation applies in any output field when you parse data with a dictionary or regular expression. Set the option in Custom mode. Use the **Select** option to select the delimiter or to add a custom delimiter. The default output delimiter is a character space.

## Maximum string length

Specifies the maximum number of characters that a parsing operation can read or write in an input or output field. You can set a maximum string length of 10,000 characters. The default maximum length is 255 characters.

# Configuring a parse asset

To create an asset in Data Quality, click **New**. When you click **New**, Data Quality prompts you to select an asset type. When you select Parse, Data Quality opens a new parse asset.

The asset displays a **Definition** tab and a **Configuration** tab. Use the Definition tab to define the name and the location of the asset. Use the Configuration tab to configure the steps that the mapping will apply to the input data.

**Note:** You can also open the Definition and Configuration tabs when you open the asset from the Explore page.

1. To create a parse asset, click **New** > **Parse**.
2. On the Definition tab, enter a name for the parse asset.
3. Optionally, enter a description.
4. Select the location in which to save the parse asset.
5. Optionally, select a data quality dimension to represent the type of data quality issue that you want the asset to examine.

   For more information about dimensions, see "Parse assets and dimensions" on page 10.
6. Save the parse asset.

   The **Asset References** panel appears. The **Asset References** panel shows additional assets that are referenced by Data Quality assets to perform transformation tasks. When you create a new asset, there are no referenced assets.
7. Optionally, add a tag to the parse asset. You can search for assets with a common tag on the Explore page.
8. Select the Configuration tab.

   Data Quality displays the configuration workspace for the parse asset. Data Quality also displays a validation message to indicate that the step configuration is incomplete.
9. Select the parsing mode to apply to your data.

   Select one of the following modes:

   - Custom. Select the custom mode to define one or more parsing steps with a dictionary or a regular expression.
   - Pre-built. Select the pre-built mode to use the built-in parsing logic in the asset.

   **Note:** Complete step 9 or step 10, based on your selection of Custom or Pre-built mode.
10. In Custom parsing mode, configure one or more steps in the asset.

    Select one or more of the following step types:

    - Regular expression. Use a regular expression to parse one or more data values in an input string to discrete outputs. You can enter a regular expression or select one from a list of built-in expressions.
    - Dictionary. Use a dictionary to parse one or more data values in an input string to one or more discrete outputs.
11. In Pre-built parsing mode, select the locale in which your input data originates and select the format in which the data is stored.

    Optionally, test the data and use the test results to enhance the pattern logic.

    You can repeat the testing process to fine-tune the pattern logic.
12. Save the asset.

For more detailed information on configuring an asset to use regular expressions, see “Configuring a regular expression step” on page 20.

For more detailed information on configuring an asset to use dictionaries, see “Configuring a dictionary step” on page 19.

For more detailed information on configuring an asset to perform pattern-based parsing, see “Configuration summary for pattern-based parsing ” on page 23.

CHAPTER 2

# Custom parsing operations

Select custom mode to define a parsing operation that uses one or more dictionary steps or regular expression steps.

You can create multiple steps in a single parse asset. At run time, the Parse transformation runs the steps that you specify.

**Note:** The Parse transformation runs any dictionary step in an asset before it runs any regular expression step. First, the transformation runs the dictionary steps in the order in which they appear in the step sequence. Next, the transformation runs the regular expression steps in the order in which they appear in the step sequence.

## Parsing with dictionaries

You can use a dictionary to find known values in an input field. Populate the dictionary with the values that you want to find.

At run time, the Parse transformation compares the values in the input field to the values in a dictionary that you select in the parse asset. The transformation writes any values that match the dictionary values to an output field.

### Example: product data

A product data set might combine data values about multiple product attributes in the same field. You can configure a parse asset to find the different values and to specify discrete fields for each type of value.

For example, you can configure the asset to split the following paint description into separate inventory elements:

```
500ML Red Matt Exterior
```

Create a step for each element that you want to find. In each step, add a dictionary that contains reference data for an element, such as size, color, style, or finish. When a mapping runs with the Parse transformation, the transformation will compare the input field values to the dictionaries that the steps specify.

The mapping will write the values to the following columns:

| Size | Color | Style | Finish |
|---|---|---|---|
| 500ML | Red | Matt | Exterior |

**Note:** The input field data may contain data values in more than one format. For example, the quantity 500ML may appear as 500 millilitres, 500ml, 0.5L, or simply 500. To capture the data in all formats, use a dictionary

that contains a column of values in each format. You can select the format that you prefer when you configure the step. Select the column that uses the preferred format as the Valid column.

# Parsing with regular expressions

You can use a regular expression to find values that match a given character structure in an input field. Create a regular expression that matches the structure of the values that you want to find. Or, select a regular expression from the list of built-in expressions in the asset.

Use a regular expression in place of a dictionary when you cannot predict the content of every value or when the range of values that you will search for is too great to add to a dictionary.

At run time, the Parse transformation applies the regular expression logic to the values in the input field. When the transformation finds a value with a structure that matches the expression logic, the transformation writes the value to the output field that the step specifies.

## Example: United States telephone numbers and Social Security numbers

A customer data set might include a column for telephone numbers. Over a period of time, many users incorrectly enter Social Security numbers into the column. You can configure a parse asset to find values that match both formats.

The following table displays the types of errors that can appear in the column:

| Value | Format |
|---|---|
| 212-555-1234 | Telephone number |
| 910-22-5555 | Social Security number |
| (518)555-8466 | Telephone number |
| (718) 555-2907 | Telephone number |
| 2125550987 | Telephone number |
| 922-823-5746 | Social Security number |
| 974-43-0202 | Social Security number |
| 212-555-3287 | Telephone number |

Create a step for each data format, and add a regular expression to each step.

For example, the parse asset contains the following built-in regular expression for United States telephone numbers:

```
1?[-. ]?\(?[0-9]{3}\)?[-. ]?[0-9]{3}[-. ]?[0-9]{4}?(?:EXT|ext|Ext|X|x|#|\.| |,)*[0-9]{3,5}|1?
[-. ]?\(?[0-9]{3}\)?[-. ]?[0-9]{3}[-. ]?[0-9]{4}
```

The asset contains the following built-in regular expression for United States Social Security numbers:

```
(.*)([0-9]{3}[- ]?[0-9]{2}[- ]?[0-9]{4})(.*)
```

Add a single output to each step for telephone numbers and Social Security numbers respectively.

# Parse asset structure

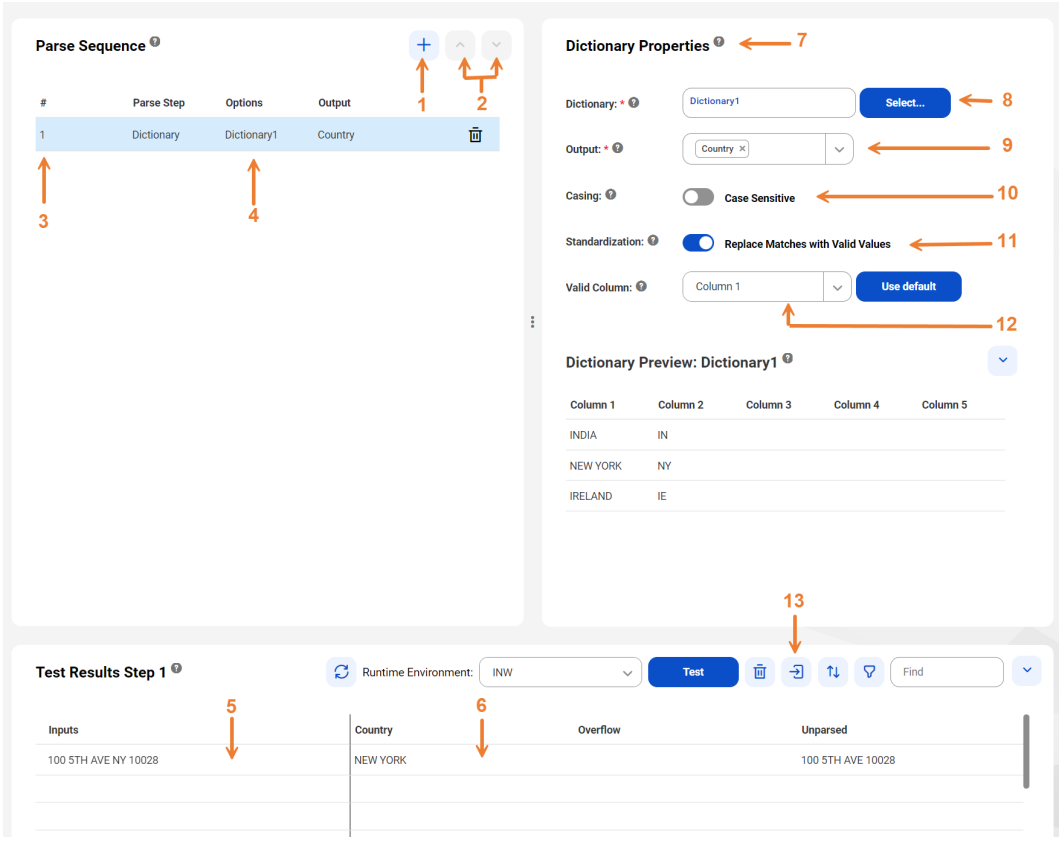
A parse asset contains options on a **Definition** tab and a **Configuration** tab. Use the Definition tab options to enter a name for the asset, optionally enter a description for the asset, and select the folder in which to store the asset.

Use the Configuration tab options to configure one or more parsing operations that a mapping will perform. You configure the parsing operations as a sequence of steps that the mapping follows at run time.

## Dictionary options on the Configuration tab

When you add a step to a parse asset, Data Quality prompts you to define a step with a regular expression or with a dictionary.

The following image shows the options that you configure when you define a step with a dictionary:




The dictionary options includes the following properties:

1. Add Step option.
   Adds a step to the asset. A step describes a parsing operation that a mapping can apply to an input data field.
2. Up and Down options.
   Moves a step that you select up or down within the step sequence.
3. Step sequence.
   Defines the order in which a mapping applies each step to the input field at run time. The mapping performs dictionary parsing operations in the order that you specify.

4. Option description.
   Identifies the dictionary that you add to the step.
5. Test input field.
   Contains the string that the Secure Agent uses to test the steps in the parse sequence.
6. Test output fields.
   Contain the result of the test. The asset displays an output field for each step and also displays output fields for overflow dat and unparsed data.
7. Step type.
   Identifies the type of step to which the properties apply. The step type can be dictionary or regular expression.
8. Dictionary name.
   Identifies the dictionary that the step applies to the input data. You select the dictionary.
9. Output name.
   An output to which the step can write each input data values that matches a dictionary value. You enter the output name. The mapping creates the output on the Parse transformation at run time.
10. Casing option.
    Specifies whether the step considers the character case of the input values that it compares to the dictionary values. For example, the character case may be relevant if you plan to parse person names from an input string, as person names begin with an uppercase letter.

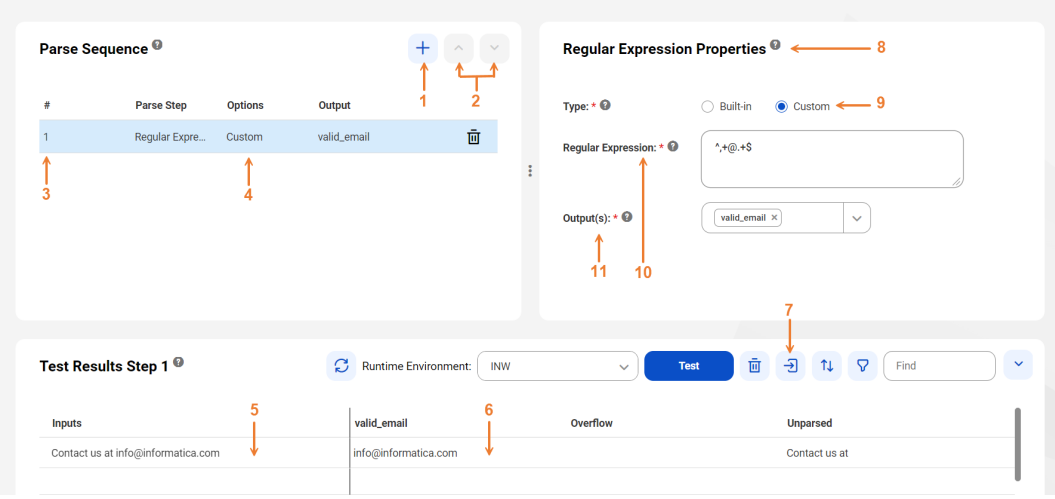
    Clear the option if the character case is not relevant to the type of data that you plan to parse from the inputs.
11. Standardization option.
    Specifies whether the step will replace any input value that matches a dictionary value with the corresponding value from the valid column in the dictionary.

    The step compares the input values to every value in the dictionary. If you clear the option, the step does not standardize any input value that matches a dictionary value.
12. Valid dictionary column name.
    Identifies the valid column in the dictionary that you select. The valid column contains the preferred versions of the values in the dictionary. The valid dictionary column is active when you select the standardization option.
13. Import file option.
    Imports the first column of data from a comma-separated file to the test panel.

# Regular expression options on the Configuration tab

When you add a step to a parse asset, Data Quality prompts you to define a step with a regular expression or with a dictionary.

The following image shows the options that you configure when you define a step with a regular expression:




The regular expression options includes the following properties:

1. Add Step option.
   Adds a step to the asset. A step describes a parsing operation that a mapping can apply to an input data field.
2. Up and Down options.
   Moves a step that you select up or down within the step sequence.
3. Step sequence.
   Defines the order in which a mapping applies each step to the input field at run time. The mapping performs regular expression parsing operations in the order that you specify.
4. Option description.
   Identifies the type of regular expression in the step. A custom regular expression is a regular expression that you define.
5. Test input field.
   Contains the string that the Secure Agent uses to test the steps in the parse sequence.
6. Test output fields.
   Contain the result of the test. The asset displays an output field for each step and also displays output fields for overflow dat and unparsed data.
7. Import file option.
   Imports the first column of data from a comma-separated file to the test panel.
8. Step type.
   Identifies the type of step to which the properties apply. The step type can be dictionary or regular expression.
9. Regular expression type.

   Specifies whether the step uses a built-in regular expression or a regular expression that you define.
10. Regular expression field.

In Custom mode, the field contains the regular expression that you specify. You enter the regular expression logic in the field.

In Built-in mode, the field displays the name of the regular expression that you select. Additional fields display the regular expression logic and the number of output fields that you must create.

11. Output name(s).
    One or more outputs to which the step can write each input data value that matches the regular expression logic. You enter the output names.

    The mapping creates the outputs on the Parse transformation at run time. If you expect an input field to contain more than one value that matches the regular expression, create an output for each value. For example, you might add two outputs if you will use the step to find data values that occur in pairs, such as map coordinates.

# Configuring a dictionary step

You can configure a step to compare the values in an input field to the values in a dictionary. The step writes matching values to one or more output fields.

1. On the **Configuration** tab, select the option to add a step to the asset.

   The **Add Step** dialog box opens onscreen.
2. Select **Dictionary** as the step type.
3. In the **Dictionary Properties** pane, select a dictionary.

   Use a dictionary that contains a range of data values that you expect the step to find at run time.

   The step compares each value in an input field to every value in the dictionary that you select, regardless of the number of columns in the dictionary.
4. In the **Outputs** field, enter the name of the field to which the step will write any input value that matches a dictionary value. The step creates the field at run time.

   Press the Enter key after you enter the field name.
5. Select or clear the **Casing** option.

   If you select the option, the step searches for values in the input fields that match both the spelling and the character case of the values in the dictionary. By default, the step ignores the character case of the values in the input fields.
6. Select or clear the **Standardization** option.

   If you select the option, the step returns the valid value from the dictionary for any input value that matches a dictionary value.
7. If you select the Standardization option, use the **Valid Column** options to identify the dictionary column that contain the valid values.

   The valid column contains the standard or preferred version of a value in cases where the dictionary contains more than one value on a given row.

   You can specify any dictionary column as the valid column within a step. To use the column that the dictionary designer set as the valid column, click **Use Default**.
8. Save the asset.

# Configuring a regular expression step

You can configure a step to apply a regular expression to the values in an input field. The step writes any value that matches the expression logic to one or more output fields.

1. On the **Configuration** tab, select the option to add a step to the asset.

   The **Add Step** dialog box opens onscreen.

2. Select **Regular Expression** as the step type.
3. Select the Built-in or Custom option.
   - To select a regular expression from a list of built-in expressions, select Built-in.
   - To enter a regular expression that the step will use, select Custom.

   Select or enter a regular expression that describes the character composition of the values that you want the step to find at run time.
4. In the **Outputs** field, enter the name of the field to which the step will write any input value that matches the regular expression logic. The step creates the field at run time.

   Press the Enter key after you enter each field name.

   You can create more than one output. Create additional outputs if the step might find more than one value that matches the expression logic within a single input field.

   A built-in regular expression might require more than one output. When you select the Built-in option, the Number of Outputs field displays the number of outputs to create.
5. Save the asset.

CHAPTER 3

# Pattern-based parsing operations

A pattern-based parsing operation reads an input string that contains multiple values and parses values to output fields based on the information that they contain. To define a pattern-based parsing operation, configure a parse asset in **Pre-built** mode.

Pattern-based parsing compares the values in the input string to a set of patterns that are built into the asset. The parsing operation selects values from the input string based on their similarity to the values in the patterns. The parsing operation additionally considers the location of each value relative to the other values that it parses from the inputs string.

You can enhance the pattern parsing logic in the asset that you configure. To enhance the logic, add one or more patterns to the asset. Use the test panel options in Pre-built mode to identify the patterns that you might add.

Parse assets support pattern-based parsing for person name data. Use pattern-based parsing to identify different types of person name data in an input string and to create an output data structure that can assign different name values to appropriate fields.

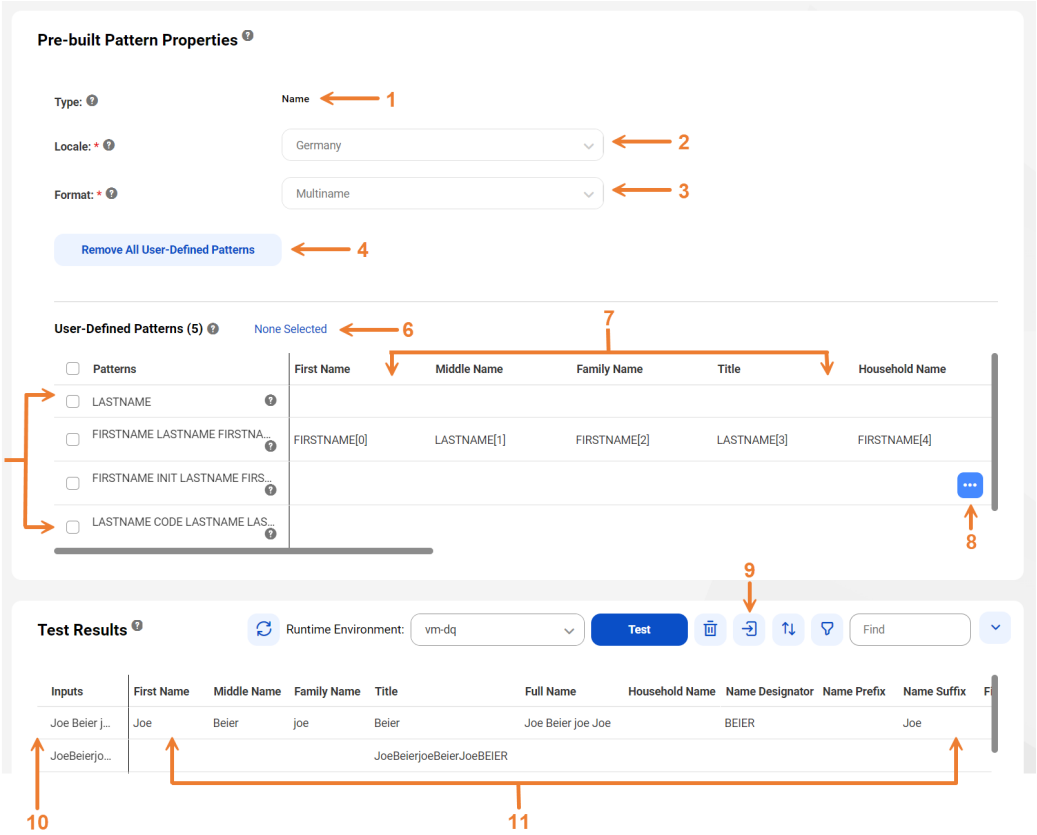
**Note:** Pattern-based parsing uses dictionaries to identify person name values and to select the appropriate output field for the values. Informatica provides the dictionaries in an asset bundle. To enable pattern-based parsing operations, install the bundle.

For more information on bundle installation, see "Installing the asset bundle for pattern-based parsing" on page 24.

## Pattern-based parsing options on the Configuration tab

When you select Pre-built mode in the Configuration tab, Data Quality prompts you to specify the criteria for person name parsing.

The following image shows the options that you configure when you define the parsing operation:




The pattern-based parsing options include the following properties:

1. Type
   Identifies the type of information that the parsing operation analyzes. Pattern-based parsing analyzes person name information.
2. Locale
   Indicates the country of origin of the data that the parsing operation analyzes.
3. Format option
   Specifies the format of the input data that the parsing operation will read. The format indicates the sequence in which the parsing operation expects to read the input data values.

   The parsing operation applies the built-in data patterns based on the format that you select.
4. Add or Remove User-Defined Patterns option
   Adds patterns that you define to the built-in patterns that the current operation uses. Or, removes all user-defined patterns from the current operation.

   The asset imports user-defined patterns from a CSV or Microsoft Excel file that you specify. You can import up to 10 MB of data from a file.
5. User-Defined Patterns
   Lists any pattern that you add to the current parsing operation.
6. Pattern selection menu
   Indicates the number of patterns that you select. You can use the menu to delete the rows or to erase the field mappings that you defined for the rows.
7. Pattern input fields

Displays the fields to which you can map the values in a user-defined pattern. Select the appropriate field for each value in any pattern that you add. The Configuration tab displays twelve possible fields.

8. Row selection menu
   Opens a set of options that you can use delete the row or to erase the field mappings that you defined for the row.

   For more information about the pattern selection and row selection options, see "Pattern selection and row selection options" on page 27.
9. Import test data option
   Imports data that the Secure Agent can test.

   The asset imports the test data from a CSV or Microsoft Excel file that you specify.
10. Test input fields
    Displays the test data that you import.
11. Test output fields
    Displays the result of any test that you run.

# Configuration summary for pattern-based parsing

To perform pattern-based parsing, create a parse asset in Data Quality and select the **Pre-built** parsing mode.

Optionally, test the asset configuration. Use the test results to update the pattern logic that the asset uses.

## Prerequisites

Before you perform pattern-based parsing, verify that the *CDQ_Name_Parsing_Reference_Data_Bundle* is present in the **Add-On Bundles** folder in Explorer.

## Testing strategies for pattern-based parsing

You can test the asset before you use it in a mapping and after you use it in a mapping. For example, you might test the asset and update the pattern data before your first mapping run. You might then use the results of the first mapping run to update the pattern data so that the mapping parses the source data more comprehensively.

When you test the asset before you run a mapping, you add one or more patterns from the test results to the pattern-based logic in the asset. When you test the asset with the results of a mapping, you can add both the patterns and the names that the patterns identify to the pattern-based logic in the asset.

## Process flow

The following steps summarize the configuration process:

1. If necessary, install the *CDQ_Name_Parsing_Reference_Data_Bundle* asset bundle.
2. Select **Pre-built** parsing mode.
3. Select the locale and the format of the data that the operation will read.

   **Note:** After you verify the locale and the data format, the asset is ready to use in a Parse transformation. You can use the steps that follow to enhance the pattern-based logic that the asset applies to your data.

   You can test and enhance the pattern-based logic before you add the asset to the Parse transformation. Or, you can run a mapping that contains the transformation and use the mapping results to update the pattern-based logic.

4. Test a sample of your data in the parse asset. Enter the values that you want to test, or import a file that contains the values.

   You can import a file and perform the test on your data in the following ways:

   - Import a file that contains name data, and run the test. You might perform this step to enhance the pattern logic before you add the asset to a transformation in a mapping.
   - Import a file that includes both names and patterns, and run the test. You might perform this step to enhance the pattern logic with the results of the mapping that you ran.

5. Review the results of the test.

   Find any name that the test failed to parse. Copy the pattern for each name to a CSV or Microsoft Excel file. Optionally, copy the name along with the pattern.

6. Import the CSV or Microsoft Excel file to the asset.
7. Map the values in each pattern to appropriate fields in the user-defined pattern grid. Then, save the asset.

   **Note:** When you map a pattern to the appropriate fields, you train the asset to recognize names with a structure that matches the pattern.

8. Run the test again and review the results.

   You may decide to import additional patterns to the asset in order to further improve the pattern parsing logic.

When you are satisfied with the performance of the parsing operation on your sample data, save the asset. A Data Integration user can add the asset to a Parse transformation in a mapping and run the mapping on your data.

## Installing the asset bundle for pattern-based parsing

Before you perform pattern-based parsing, verify that the dictionaries that the parsing operation reads are installed for your organization. Informatica provides the dictionaries in a bundle named *CDQ_Name_Parsing_Reference_Data_Bundle*. Find the bundle in the **Add-On Bundles** folder on the Explore page.

If the bundle is not present, install the bundle. You install the bundle in the Administrator service.

**Note:** *CDQ_Name_Parsing_Reference_Data_Bundle* contains 84 dictionaries. If you find the bundle on the Explore page and the bundle contents are incomplete, uninstall the bundle from the **Add-On Bundles** page in Administrator. Then, install the bundle.

To install a bundle, perform the following steps:

1. In Administrator, select **Add-On Bundles**.
2. Click **Available Bundles**.

   The Available Bundles tab lists the public and private bundles that are available for installation or copying.

3. If the bundle that you want to install is an unlisted bundle, enter the bundle access code in the **Find** field.
4. Click the bundle name to open the Bundle Details page.
5. Verify that the **Allow** field is set to **Reference** or to **Reference and Copy**.

   You cannot install a bundle that is configured for copying only.

6. Click **Install**.

Administrator displays a notification to indicate the status of the installation.

You can find the installed bundle name on the **Installed Bundles** tab of the **Add-On Bundles** page in Administrator. In Data Quality, the bundle is added to the **Add-On Bundles** project in the Explore page.

# Configuring the pre-built options for pattern-based parsing

If your data matches the default settings in **Pre-built** mode on the parse asset, you can save the asset for use in a Parse transformation with minimal configuration.

1. On the **Configuration** tab, select the **Pre-built** parsing mode.
2. In the **Pre-built Pattern Properties** pane, configure the following options:
   - Locale. Specifies the country to which the input data belongs. The configuration specifies the United States by default.

     **Note:** Each locale has a unique set of output fields.
   - Format. Specifies the format of the data that the operation will read. The default format is *Last Name, First Name, Middle Name*.
3. Save the asset.

After you complete the configuration steps, test the asset configuration.

# Testing and updating the pattern-based parsing configuration

Test a parse asset to verify that data flows through the asset in the ways that you expect. You can then update the asset with the pattern for any name that the test failed to parse. The asset stores the patterns that you add and includes them in the pattern logic that the Parse transformation uses at run time.

You can update the asset with pattern data, and you can update the asset both with patterns and with the names in your input data that are associated with the patterns.

You might update an asset exclusively with pattern data when you test the asset before you add it to a Parse transformation. You might update an asset with both pattern data and name data after you run a mapping with the Parse transformation and review the output from the transformation.

## Iterative testing

Testing the asset and updating the pattern-based logic for a source data set can be an iterative process. You might test the asset first with a sample of your source data and later test the asset with the name data and associated patterns that you read from the transformation output data.

To test the asset with name data only, see "Testing and updating the pattern-based parsing configuration" on page 25.

To test the asset with patterns and name data, see "Testing the parsing configuration with pattern and name data" on page 26.

## Testing the parsing configuration with name data

The following steps describe the process to test the parse asset with name data that you import. When you test the data, you might find names that the asset does not parse. You can then add the patterns for the unparsed names to the asset logic.

1. Open the parse asset that you created for pattern-based parsing.
2. Select the **Configuration** tab.
3. Select a Secure Agent to run the test.

   To refresh the list of active Secure Agents, click the **Refresh** icon.

4. Import a sample of the source data that the Parse transformation will run on.

   Use the Import option in the **Test Results** pane to import the data from a CSV or Microsoft Excel file. You can import up to 10 MB of data from a file. The sample data must populate the first column in the file.

   The input data appears in the **Inputs** column.
5. Click **Test**.

   The output columns display the test results.
6. Review the results of the test:
   - Find any input name that the test did not parse. Look for input names in the **Unparsed Data** field.
   - If you find a name that did not parse, make a copy of the pattern that the test assigned to the name.

     The test writes the pattern for each name to the **Labeled Name** field. Copy the pattern from the Labeled Name field to a CSV or Microsoft Excel file.
7. Import the file that contains the pattern to the asset.

   Use the **Add User-Defined Patterns** option to import the data. The patterns must populate the first column in the file.
8. Map the values in each pattern to an appropriate field in the pattern grid. For each value, select the field that best matches the type of information that the value represents.

   Use the number in each pattern value as a guide when you map the values. The numbers match the order in which the data values appear in the input row. The numbered values in each pattern begin at (0).
9. After you map the pattern values to the appropriate fields, run the test again and review the results.

   You may decide to import additional patterns to further update the pattern parsing logic.
10. When you are satisfied with the performance of the parsing operation on your sample data, save the asset.

## Testing the parsing configuration with pattern and name data

The following steps describe the process to test the parse asset with pattern data and associated name data. For example, you might discover names that the Parse transformation did not parse successfully at run time. Import the names and the associated patterns from the transformation output to the asset in Data Quality.

To import the names and associated patterns, copy the data to a CSV or Microsoft Excel file. The pattern values must populate the first column in the file, and the name values must populate the second column.

1. Open the parse asset that you created for pattern-based parsing.
2. Select the **Configuration** tab.
3. Select a runtime environment in which to test the configuration.
4. Use the **Add User-Defined Patterns** option to import the unparsed name and pattern data that the Parse transformation wrote as output.

   When you select the file that contains the data, the **Import Patterns** dialog box opens. Select the **Import Input Data** option in the dialog box.

   The pattern data appears in the **User-Defined Patterns** pane and the name data appears in the Inputs column of the **Test Results** pane.
5. Map the values in each pattern to an appropriate field in the pattern grid. For each value, select the field that best matches the type of information that the value represents.

   Use the number in each pattern value as a guide when you map the values. The numbers match the order in which the data values appear in the input row. The numbered values in each pattern begin at (0).
6. Click **Test**.

The output columns display the test results.

7. Review the results of the test:
   - Find any input name that the test did not parse. Look for input names in the **Unparsed Data** field.
   - If you find a name that did not parse, make a copy of the pattern that the test assigned to the name. The test writes the pattern for each name to the **Labeled Name** field.

     You can copy the patterns from the Labeled Name field to the file that you previously imported, or you can copy the patterns to a new file. Bear in mind that when you import data to the asset, you erase the prior test and pattern data on the Configuration tab.
8. Import the file that contains the latest pattern and name data.
9. Map the pattern values to the appropriate fields in the pattern grid.
10. Run the test again and review the results.

    You may decide to import additional patterns to further update the pattern parsing logic.
11. When you are satisfied with the performance of the parsing operation on your sample data, save the asset.

**Further information**

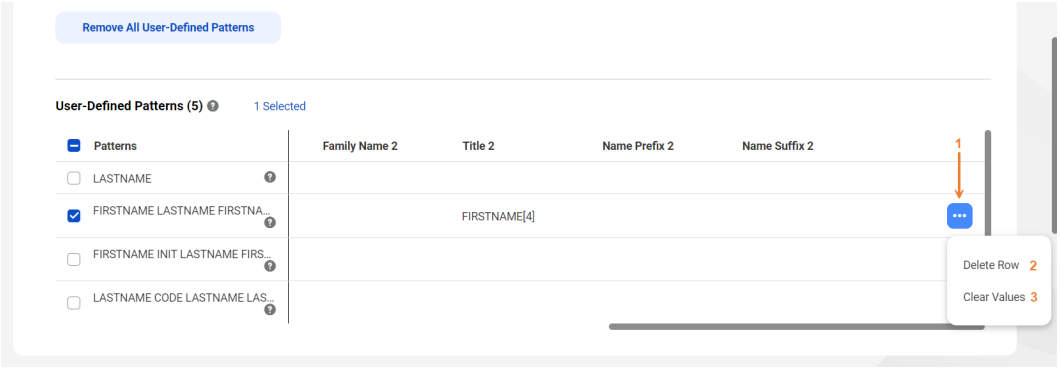
For more information about importing files, see "Rules and guidelines for test data" on page 32.

For more information about the meaning of the pattern label values, see "Rules and guidelines for user-defined pattern labels" on page 28.

## Pattern selection and row selection options

When you import data to the **User--Defined Patterns** list, you can use the pattern and row selection options to undo any pattern match that you define and to delete unwanted rows of data from the list. Use the row selection options to clear the pattern matches from a single row or delete a single row. Use the pattern selection options to clear the pattern matches from multiple rows or delete multiple rows in a single action.

The following image shows the row selection options:



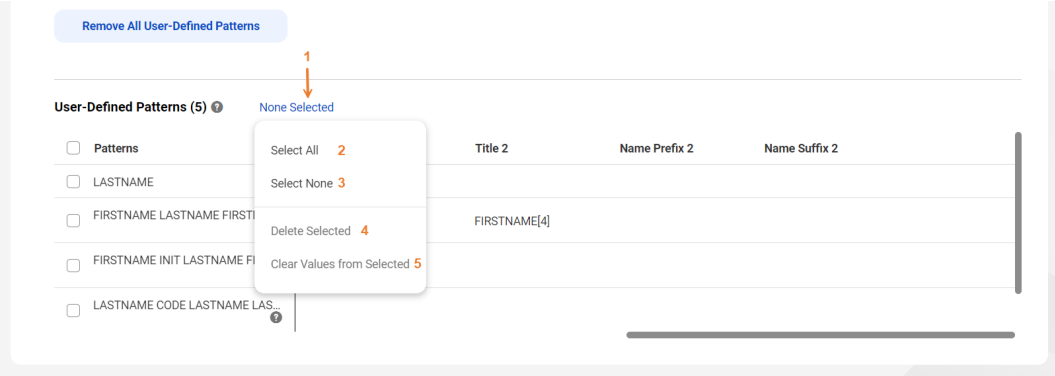

The row selection options include the following properties:

1. Row properties option
   Opens the menu of row selection options.
2. Delete Row
   Deletes the current row of data.

   Clear Values

   Removes all pattern matches that you selected in the current row. The option does not delete the pattern that you imported.

The following image shows the pattern selection options:




The pattern selection options include the following properties:

1. Pattern properties option
   Opens the menu of pattern selection options. The option also displays the number of rows that you selected.
2. Select All
   Selects all rows under User-Defined Patterns.
3. Select None
   Delesects all rows under User-Defined Patterns.

   **Note:** You can also use the check boxes beside each row to select or deselect one or more rows.
4. Delete Selected
   Deletes the rows of data that you selected.
5. Clear Values from Selected
   Removes all pattern matches from the rows that you selected. The option does not delete the patterns that you imported.

# Rules and guidelines for user-defined pattern labels

When you create a user-defined pattern, you must use labels that the parse asset recognizes.

Considere the following rules and guidelines when you add labels to a user-defined pattern:

- Use label values that appear in the **Labeled Name** field in the test panel.
- The following table describes the labels that the parse asset recognizes:

| Label | Description |
|---|---|
| FIRSTNAME | First name of a person. |
| LASTNAME | Family name. |
| MIDDLENAME | Middle name of a person. |
| LN_ID | Affix to family name. For example, Mac, Van, al. |

| Label | Description |
|---|---|
| INIT | Initial, or single alphabetic character. |
| TITLE | Occupation name. |
| SUFFIX | Person name suffix, such as a professional qualification. |
| INITSET | Two or more initials separated by spaces or periods. |
| PREFIX | Person name prefix. For example, DR, MISS. |
| DESIGNATOR | Designated recipient identifier. For example: CARE OF. |
| HOUSEHOLD | Household identifier. For example: Current Resident. |
| WORD | Alphabetic string. |
| WORDSYMBOL | Combination of one or more alphabetic characters and punctuation symbols. |
| CONNECTOR | Conjunction. For example: and, or. |
| CODE | Alphanumeric string. |
| NUMBER | Numeric string. |
| SYMBOL | Punctuation symbol. |
| UNDEFINED | Undefined string. |
| CONNECT | Conjunction. For example: and, or. |
| dq_dash | Dash symbol in a Spanish name. |
| dq_comma | Comma symbol in a Spanish name. |
| dq_quote | Quote symbol in a Spanish name. |
| dummy | Value that is not associated with a name. e.g. Generico. |
| found | Portuguese name prefix. |
| COMMA | Comma symbol. |
| #DASH | Dash symbol. |
| #FNDOLN | Family name that begins with [O']. |
| #!FND | Family name affix [O'] without an apostrophe. |
| amb | Ambiguous Spanish name. May be first or family name. |
| apell | Spanish family name. |

| Label | Description |
| --- | --- |
| nom | Spanish first name. |
| inicial | Initial letter in a Spanish name. |

CHAPTER 4

# Test parse assets

Test the asset in Data Quality to verify that the asset logic generates the results that you expect.

**Note:** The steps to test a parse asset in pattern-based parsing mode are integral to the process flow for pattern-based parsing. To read more about the steps test the asset in pattern-based parsing mode, see "Testing and updating the pattern-based parsing configuration" on page 25.

## Testing a parse asset in Custom mode

Test a parse asset to verify that the data flows through the asset in the ways that you expect.

1. Open the parse asset.
2. Select the **Configuration** tab.
3. Select a step in the step sequence.

   You will test the logic in the sequence up to the step that you select. To test the logic in the complete asset, select the final step in the sequence.
4. Select a Secure Agent to run the test.

   To refresh the list of active Secure Agents, click the **Refresh** icon.
5. Enter one or more data values in the input column, or import the data to test. To import the data, click the **Import** option in the test panel.

   For more information on importing data, see "Rules and guidelines for test data" on page 32.
6. Click **Test**.

   The test runs the steps in the asset in the following sequence:

   - The test runs all dictionary steps in the order in which they appear in the step sequence.
   - The test runs all regular expression steps in the order in which they appear in the step sequence.

   The output columns display the test results.
7. Verify that the steps read the input values and write the output values that you expect.

# Rules and guidelines for test data

You can import data to the test panel in the parse asset and save the test data in the asset configuration.

Consider the following rules and guidelines when you add data to the test panel in a parse asset:

- The import option supports CSV and Microsoft Excel files.
- You can import up to 200 consecutive rows of data from a delimited file. You can specify the row at which the import starts.

  **Note:** Before you import, check the file for column headings. If the first row in the import file contains column headings, start the import at line 2 or lower.
- The Maximum string length property applies to the data that you import or enter in the test panel. Import or enter a string on each row that does not exceed the value on the property.
- If you import a CSV file that contains multiple columns or uses a text qualifier, verify that the file uses a delimiter or a text qualifier that the Secure Agent recognizes. By default, the *Comma* option is the delimiter for the column data. By default, the *No quotes* option is the text qualifier for the data. You can update the delimiter and text qualifier characters when you select the data to import. The Delimiter and Text Qualifier options are not required when you import a Microsoft Excel file.
- The Secure Agent saves the data that you import to the asset when you save the asset. If you change an option in the asset configuration, you may lose any unsaved test data.

# INDEX


## C

Cloud Application Integration community
  URL 5
Cloud Developer community
  URL 5

## D

Data Integration community
  URL 5

## I

Informatica Global Customer Support
  contact information 6
Informatica Intelligent Cloud Services
  web site 5

## M

maintenance outages 6

## P

parse assets
  and dimensions 10
parse assets *(continued)*
  and mappings 9
  creating a parse asset 12
  parsing modes 7
  testing a parse asset 31

## S

status
  Informatica Intelligent Cloud Services 6
system status 6

## T

trust site
  description 6

## U

upgrade notifications 6

## W

web site 5