Informatica

Informatica® Cloud Data Integration
Spring 2020 April

Components

Informatica Cloud Data Integration Components
Spring 2020 April
August 2020

Publication Date: 2020-09-04

Table of Contents

Preface

Use *Components* to learn about reusable assets that you can create in Data Integration. Learn how to create components and add them to transformations, mappings, and tasks.

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. This site contains information about Informatica Cloud integration services.

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/

Data Integration connector documentation

You can access documentation for Data Integration Connectors at the Documentation Portal. To explore the Documentation Portal, visit https://docs.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, go to https://status.informatica.com/ and click **SUBSCRIBE TO UPDATES**. You can then 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 Customer Support Center by telephone or online.

For online support, click **Submit Support Request** in Informatica Intelligent Cloud Services. You can also use Online Support to log a case. Online Support requires a login. You can request a login at https://network.informatica.com/welcome.

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

Components

Components are assets that support mappings and tasks.

You can use the following components:

Business services

Define a business service to use in Web Service transformations.

For more information about business services, see Chapter 2, “Business services” on page 10.

File listeners

Create a file listener that listens to files on a specific location and notifies subscribers when files arrive at the location and when files in the location are updated or deleted.

For more information about file listeners, see Chapter 3, “File listeners” on page 12.

Fixed-width file formats

Configure reusable formats that you can use for fixed-width flat file sources and targets to enhance readability.

For more information about fixed-width file formats, see Chapter 4, “Fixed-width file formats” on page 20.

Hierarchical schemas

Upload an XML schema, XML sample file, or JSON sample file to use in mappings that include a Hierarchy transformation.

For more information about hierarchical schemas, see Chapter 5, “Hierarchical schemas” on page 24.

Intelligent structure models

Create or export a model for the following use cases:

- Create a model for semi-structured or unstructured data to use in a Structure Parser transformation in a mapping.
- Export a model to use with mappings in Informatica Big Data Management.
- Export a model to process partner messages in Cloud B2B Gateway.

For more information about intelligent structure models, see Chapter 6, “Intelligent structure models” on page 25.

Mapplets

Define reusable transformation logic to use in Mapplet transformations in one of the following ways:

- Create a mapplet in Data Integration.
- Upload a PowerCenter mapplet XML file or an SAP mapplet to extend transformation logic in synchronization tasks and mappings.

For more information about mapplets, see Chapter 7, “Mapplets” on page 45.

Saved queries

Create reusable custom queries that you can use as a source in synchronization tasks.

For more information about saved queries, see Chapter 8, “Saved queries” on page 52.

Shared sequence

Define a reusable sequence to use in multiple Sequence Generator transformations.

For more information about shared sequences, see Chapter 9, “Shared sequences” on page 56.

User-defined functions

Create reusable functions that you can use in transformation expressions and in other user-defined functions.

For more information about user-defined functions, see Chapter 10, “User-defined functions” on page 59.

Visio templates

Design transformation logic in Visio and then upload the template and configure the parameters to use in mapping tasks.

For more information about Visio templates, see *Mappings*.

Create components on the **New Asset** page.

CHAPTER 2

Business services

A business service is a web service with configured operations. Define a business service to add operations to the Web Services transformation in the Mapping Designer.

Define business services and store them in your project folders. You can use business service definitions in multiple mappings.

Defining a business service

To define a business service, perform the following steps:

1. Click **New** > **Components** > **Business Services** and then click **Create**.
2. Enter the business service details and select the Web Services Consumer connection.
3. Select the operation you want to use from the web service.
4. If necessary, configure the operation to specify the choice elements and derived type elements for the request and the response.

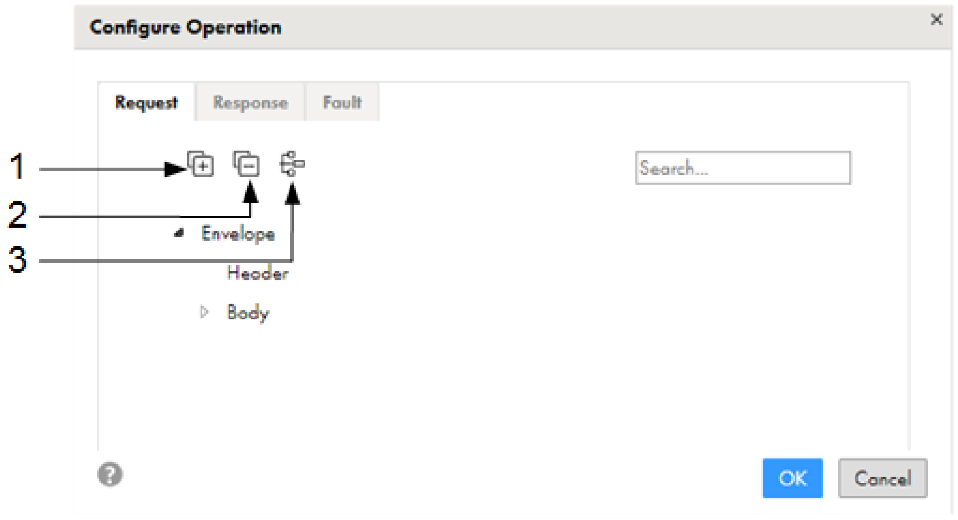
   If operation components include choice elements or complexType elements where the abstract attribute is true, then you must choose one or more elements or derived types when you configure the operation mapping.

   Optionally, for a complexType element where the abstract attribute is false, you can also select a derived type for a complexType element.

   a. For the operation you want to configure, click **Configure**.
   b. From the **Configure Operation** window, click the **Request**, **Response**, or **Fault** tab and navigate to the node you need to configure.

      Note: If the WSDL uses the anyAttribute element, the element will not appear for the request or the response.

You can click the icons at the top to navigate to the nodes that you need to configure:



1. Expand all nodes.
2. Collapse all nodes.
3. Show only nodes that you must configure. If you select this icon and no nodes expand, then you do not have to choose any choice elements or derived types.

c. Select the choice elements or derived types.

   Ensure that you configure the request and the response for the operation by clicking the relevant tabs.

d. Save the configured operation.

5. Optionally, add more operations and configure the operations if necessary.
6. Save the business service.

   If you have not configured all the required choice elements and derived types, then you cannot save the business service.

CHAPTER 3

File listeners

A file listener listens to files on a defined location. Taskflows, mass ingestion tasks, and B2B Gateway partner flows use file listeners to monitor specific folders, and receive notification through a call-back API when a file event occurs. A file event occurs when new files arrive to the monitored folder or the files in the monitored folder are updated or deleted.

You define a file listener that listens to a specific folder and the file pattern. You can define the file events that trigger a notification to the following assets:

- Taskflows
- Mass ingestion tasks
- B2B Gateway partners

For example, you can define whether mass ingestion tasks are notified when new files arrive or files in the monitored location are updated or deleted. You then assign the file listener to a mass ingestion task. The file listener listens to source folder, and notifies the mass ingestion task when a file arrived, file updated, or file deleted event occurs. You can configure the source as a server event or a connector. The mass ingestion task then runs and picks up files from the source folder.

A file listener sends notifications to the user who created it when it starts listening to the folder, when it stops listening to the folder, and if errors occur while it listens to the folder.

The file listener creates a job when the file listener starts and lists the job instance in the file transfer logs page. The file listener updates the job when files events occur. The file listener job details appear in the file listener job properties.

A file listener that is not used by a taskflow, mass ingestion task, or B2B Gateway partner is not active and does not listen to the defined folder.

File listeners in mass ingestion tasks

You can use a file listener as a source and to schedule file monitors in mass ingestion tasks.

In mass ingestion tasks with the following source types, you can schedule the task to run when it receives notifications from a file listener:

- Local folder
- Advanced FTP V2
- Advanced SFTP V2
- Advanced FTPS V2
- Amazon S3 V2

- Microsoft Azure Data Lake Store Gen2
- Microsoft Azure Data Lake Store V3
- Microsoft Azure Blob Storage V3
- Google Cloud Storage V2
- Hadoop Distributed File Storage (HDFS) V2

File listeners in taskflows

You can use a file listener in a taskflow when the file listener is configured to listen to connections.

You can use a file listener in a taskflow for the following use cases:

To invoke a taskflow through a file listener

You can invoke a taskflow through a file listener with the connector source type.

Within the taskflow, define the binding type as **Event** and select the file listener as the event source. When you publish the taskflow, the taskflow subscribes to the file listener that is defined in it. When a file event occurs, the file listener invokes the taskflow.

For example, if you configure the file listener to listen for new files on a folder, the file listener invokes the associated taskflow each time a new file arrives in the specified folder.

To orchestrate taskflow execution through file events

You can orchestrate taskflow execution through file events by using the File Watch Task step in a taskflow.

You can add a File Watch Task step to a taskflow to listen to files in a defined location and monitor file events. In the File Watch Task step, you can select an existing file listener with the connector source type. You can use file events to orchestrate taskflow execution.

For example, you can wait for a file to arrive at a particular location and then consume the file in a subsequent step.

File listeners in B2B Gateway partner flows

You can use a file listener to trigger B2B Gateway partner inbound and outbound flows.

When you configure the partner, you select the file listener to trigger the inbound or outbound flow. When you save the partner, the partner subscribes to the file listener. The file listener triggers the flow according to the rules defined in the file listener.

For example, you configure a file listener to listen for new files arriving in a folder and then configure the partner to use the file listener for inbound flows. When the partner puts files into the specified folder, the file listener triggers the inbound flow.

Behavioral differences in file listeners

Certain behavioral differences exist when a file listener is used in mass ingestion tasks and B2B Gateway partner flows, and in the File Watch Task step of a taskflow.

The following table describes the behavioral differences in file listeners based on where they are used:

Behavior	File listener used in mass ingestion tasks and B2B Gateway partner flows	File listener used in the File Watch Task step of a taskflow
Lifecycle	The file listener runs until the first occurrence of a file event or until the configured end time.	The file listener runs until the first occurrence of a file event or until a timeout occurs. If a file event does not occur, by default, the taskflow waits for 5 minutes or for the overridden value defined in the **Time Out** input field of the File Watch Task step. The maximum time period for which the taskflow waits is 7 days, after which it times out.
End time of run	The latest end time of the run is at 11:55 PM on the configured end date and time zone, and cannot extend to the following day.	The file listener runs until a file event or timeout occurs, and does not depend on any end date or time zone.
Snapshots	- All the file listener instances share the same snapshot. - Snapshots are never deleted. - The file listener run is listed in the **File Transfer Logs** tab of the Monitor service with the file listener name as the instance name. For example, if the file listener name is **FL_Delete**, the instance name that you need to look for in the **File Transfer Logs** tab of the Monitor service would be **FL_Delete**.	- Each file listener instance maintains its own snapshot. - Snapshots are deleted immediately after the jobs complete. - Each file listener run is listed in the **File Transfer Logs** tab of the Monitor service. Append the monitorJobId that you see in the output fields to the file listener name to find the instance name in the **File Transfer Logs** tab of the Monitor service. For example, if the monitorJobId is **7500** and the name of the file listener is **FL_Arrive**, the instance name that you need to look for in the **File Transfer Logs** tab of the Monitor service would be **FL_Arrive-7500**.
Start and Stop	You can start and stop a file listener instance from Data Integration or by using the File Transfer REST API resources.	When you run a taskflow that contains a File Watch Task step, the associated file listener instance starts. The file listener instance stops when the first occurrence of a file event or a timeout occurs. When you manually start or stop a file listener instance, the taskflow is not impacted.

Configuring a file listener

Before you create file listeners, verify that the following conditions exist:

- The organization has the appropriate licenses.
- The file listener is running on the secure agent.

Perform the following steps to configure a file listener:

1. To create a file listener, click **New** > **Components** > **File Listener**, and then click **Create**.

   To edit a file listener, on the **Explore** page, navigate to the file listener. In the row that lists the file listener, click **Actions** and select **Edit**.

2. Configure the following file listener details:

Parameter	Description
File Listener Name	Name of the file listener. A file listener name must be unique within the organization. File listener names can contain alphanumeric characters, spaces, and underscores. The names must begin with an alphabetic character or underscore. File listener names are not case sensitive.
Location	Project directory in which the file listener is created.
Description	Optional description of the file listener. Maximum length is 1024 characters.
Status	Status of the file listener: Enabled or Disabled. A disabled file listener does not listen to files on the designated folder.
Runtime Environment	Runtime environment that contains the Secure Agent used to run the file listener.
Source Type	Type of source to which file listener listens. Select one of the following source types: - **Server**. File listener listens to the server. For more information about parameters and file listener rules that you configure for the server, see "Configuring file listener for a server source type" on page 16. - **Connector**. File listener listens to the connection. For more information about parameters and file listener rules that you configure for the connector, see "Configuring file listener for a connector source type" on page 17.

3. Configure the schedule by which the file listener runs:

Parameter	Description
Run	Frequency at which the file listener runs, daily, weekly, or monthly.
Start Date	Date on which the file listener starts running.
End Date	Date until which the file listener runs.
Repeat indefinitely	The file listener runs without an end date.
Start At	Time of day when the file listener starts running.
Check Until	Time of day when the file listener stops running.

Parameter	Description
Check Every	Frequency at which the file listener checks for files in the folder to which it listens, by seconds, minutes, or hours.
Time Zone	Time zone according to which the file listener runs.
Days to Run	Days of the week when the file listener runs when you select to run the file listener weekly.
Day of Month	Day of the month when the file listener runs when you select to run the file listener every month.
Day of Week	Day of the week the file listener runs when you select to run the file listener every month
Week	The week the file listener runs when you select to run the file listener every month. For example, to run the file listener on the fourth Sunday of the month, select **Day of Week**: **Sunday** and **Week**: **Fourth**.

4. Click **Save**.

Configuring file listener for a server source type

When you select the source type as server in the file listener details, file listener listens to the server events.

You can configure a file listener to listen to events on AS2, HTTPS, and SFTP servers. For example, you can configure the file listener to send notification when the AS2 server receives a file. For more information about configuring file listener details, see “Configuring a file listener” on page 14.

Configure the following parameters and the file listener rules when you select a server source type:

1. Configure the following parameters to define the events that file listener listens:

Parameter	Description
Event Provider	The server type to which the file listener listens. You can configure the file listener to listen to the AS2, HTTPS, or the SFTP server.
Event Type	The type of server event to which file listener listens. The event types that you can define for the AS2 server are: - AS2 Message Receive Failed - AS2 Message Receive Successful The event types that you can define for the HTTPS server are: - HTTPS Upload Failed - HTTPS Upload Successful The event types that you can define for the SFTP server are: - SFTP Upload Failed - SFTP Upload Successful For example, if the event type is SFTP Upload Failed, then file listener sends notifications when the file upload fails on the SFTP server.

2. Configure the following parameters to define listener rules:

Parameter	Description
Key	The event attributes.
Type	The methods to filter the key value. - Contains. Filters keys that contain the value. - Equals. Finds an exact match to run the rule.
Value	Value of the key.

Configuring file listener for a connector source type

When you select the source type as connector in the file listener details, the file listener listens to the connection.

For more information about configuring file listener details, see "Configuring a file listener" on page 14.

1. Configure the following parameters to define the connection:

Parameter	Description
Connection Type	Type of the connection to which the file listener listens.
Connection	Connection to which the file listener listens.

2. Configure the following parameters to define listener rules:

Parameter	Description
Folder Path	Path to the folder to which the file listener listens.
Pattern Type	Determines the pattern of the file name to which the file listener listens. Select one of the following patterns: - Wildcard - Regex
File Pattern	File name pattern to which the file listener listens. - Wildcard. Use wildcard patterns of file name. - Regex. Use a regular expression to match the file name pattern. If you select regex pattern type, consider the following examples: - Use the following syntax to listen to all files except for files whose name contains out, foo, and baz. `^(?!.*(?:out|baz|foo)).*$ all except` - Use the following syntax to listen to all files that have an extension of doc,docx, pdf. `([a-zA-Z0-9\s_\\.\-\(\):])+(.doc|.docx|.pdf)$` - Use the following syntax to listen to all text file except for files whose name contains `out.txt`. `^(?!out).*\.txt$`
Check for files in sub-folders	Indicates whether the file listener checks for files in sub-folders under the folder to which it listens.

Parameter	Description
Notify when file	Determines when the file listener must send notifications to the services that are registered to it: - Arrived. Sends notifications when files arrive at the folder to which the file listener listens. - Updated. Sends notifications when files in the folder to which the file listener listens are updated. - Deleted. Sends notifications when files in the folder to which the file listener listens are deleted. You can select as many options as required.
Stop checking if rules are met	The file listener stops listening to the folder when the listener rules are met. For example, if you configure the file listener to send notifications when the files in the folder to which it listens are deleted, the listener stops listening to the folder when the first event of file deletion occurs in the folder. If this option is not selected, the file listener notifies the registered application on events and continues to listen for subsequent events.
Check File stability	The file listener verifies that the entire file is copied to the folder to which it listens before notifying the registered services. **Tip:** Select this option if you transfer large files, where the process of writing the files to the folder is not instant.
Notify if files exist on first run	When the file listener runs for the first time, it sends a notification if files exist in the folder to which it listens.

Start and stop a file listener manually

A file listener runs at a defined frequency according to the run schedule. If you do not want to run the file listener on a pre defined schedule, you can start or stop a file listener manually.

A file listener listens to files on a defined location from the start time until the end time when you schedule a file listener run.

Consider the following file listener behavior when you start and stop a file listener manually:

- You start a file listener before a scheduled run. The file listener that is triggered manually runs and stops at the start time of the schedule run.
- You start a file listener after the end time of a schedule run. The file listener runs once and stops.
- You cannot start a file listener manually when a file listener is already running according to the run schedule.
- When the agent version is upgraded, the file listener is automatically submitted to the new agent version.
- If the agent fails, such as a temporary network disruption, the file listener is automatically submitted to the agent when the agent restarts. Events such as Arrive, Update, and Delete that were submitted when the agent failed are notified by the file listener when the agent restarts.
- You stop an existing file listener run. The file listener stops listening to files for that run. A new run starts at the time of the next schedule time unless the file listener is started manually.

Starting and stopping a file listener

Perform the following steps to start or stop a file listener manually.

1. Select an existing file listener or create a new file listener.

   The **Start** button is enabled.

2. Click **Start**.

   The file listener validates the configuration and starts listening to files defined in the file listener configuration rules. The file listener stops after the first run.

3. Click **Stop** to stop the file listener.

CHAPTER 4

Fixed-width file formats

You can create and save fixed-width file formats that specify the formatting details for fixed-width flat files.

You can use a fixed-width flat file as a source or target in mappings and mapping tasks. When you select a flat file connection for a Source transformation or Target transformation, you specify the flat file type. If you select the fixed-width flat file type, you select the most appropriate fixed-width file format to use based on the data in the fixed-width flat file.

You can create multiple fixed-width file formats. For example, you might have a fixed-width file format to use for fixed-width flat files that contain quarterly sales data and another fixed-width file format to use for fixed-width flat files that contain inventory data.

If a fixed-width file format does not already exist, you must create a fixed-width file format before you create a mapping or mapping task that uses a fixed-width flat file.

To configure a fixed-width file format, you specify the number of columns and the width, name, and datatype for each column. You can also set advanced fixed-width format properties. For example, you can specify how to handle null characters or specify the default date format for each column.

You can delete fixed-width file formats that you no longer use. You cannot delete a fixed-width file format that is used in a mapping or mapping task.

Creating a fixed-width file format

Create a fixed-width file format so that you can use a fixed-width flat file as a source or target in a mapping.

When you specify format options for a fixed-width file format, use a sample flat file. The data in the sample file appears on the page to help you determine the appropriate format options for your data. The format options that you specify do not alter or save the sample file.

1. Click **New** > **Components** > **Fixed Width File Format** and then click **Create**.

   To edit a fixed-width file format, on the **Explore** page, navigate to the fixed-width file format. In the row that contains the fixed-width file format, click **Actions** and select **Edit**.

2. Enter the following fixed-width file format details:

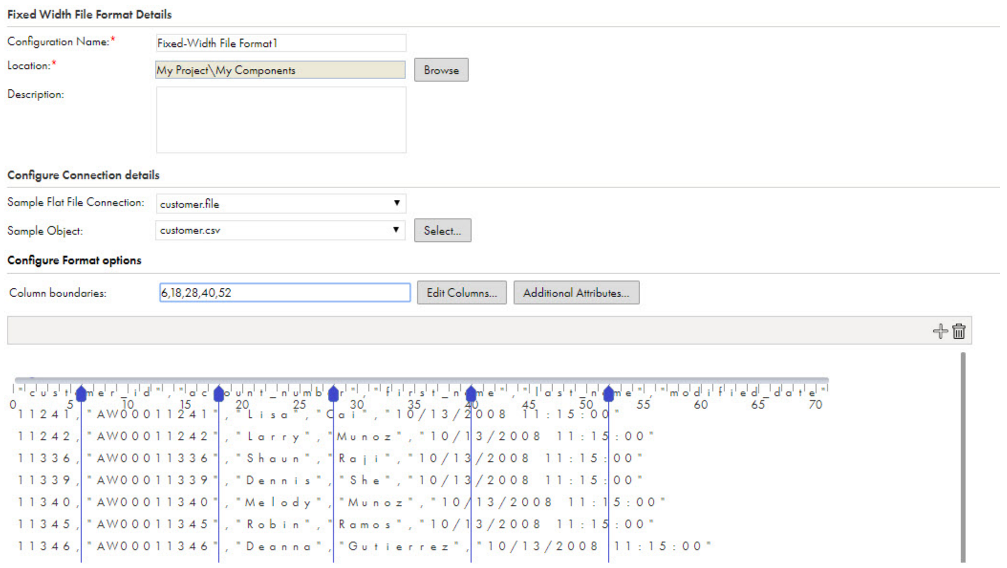
Property	Description
Configuration Name	Name of the fixed-width file format.
Location	Location of the fixed-width file format. Browse to the folder where you want to store the fixed-width file format or use the default location. If the **Explore** page is currently active and a project or folder is selected, the default location for the asset is the selected project or folder. Otherwise, the default location is the location of the most recently saved asset.
Description	Description of the fixed-width file format.

3. Specify the following connection details for the sample fiat file:

Property	Description
Sample Flat File Connection	Connection to the sample file that you use to specify the fixed-width format.
Sample Object	Sample file that you use to specify the format options.

4. Configure the column boundaries to fit the data in your sample file. You can enter the boundaries in the **Column boundaries** field or you can position the boundaries using the ruler.
    - To configure column boundaries in the **Column boundaries** field, enter the number of bytes to determine the width of the first column. Then enter boundaries for the subsequent columns. Separate each value with a comma. Press Enter after you enter a boundary to see the placement of the boundary on the ruler.
    - To configure column boundaries using the ruler, use the mouse to drag the boundary to the appropriate location on the ruler for the first column. Each mark on the ruler represents a byte. To add new boundaries, click the ruler or click **Add**. Use the mouse to drag boundaries to the appropriate position on the ruler as necessary.

The following image shows a fixed-width file format with boundaries configured for six columns:



Fixed Width File Format Details

Configuration Name:* Fixed-Width File Format1

Location:* My Project\My Components

Description:

Configure Connection details

Sample Flat File Connection: customer.file

Sample Object: customer.csv

Configure Format options

Column boundaries: 6,18,28,40,52

```
"customer_id","account_number","first_name","last_name","modified_date"
0    5    10   15   20   25   30   35   40   45   50   55   60   65   70
11241,"AW00011241","Lisa","Cai","10/13/2008 11:15:00"
11242,"AW00011242","Larry","Munoz","10/13/2008 11:15:00"
11336,"AW00011336","Shaun","Raji","10/13/2008 11:15:00"
11339,"AW00011339","Dennis","She","10/13/2008 11:15:00"
11340,"AW00011340","Melody","Munoz","10/13/2008 11:15:00"
11345,"AW00011345","Robin","Ramos","10/13/2008 11:15:00"
11346,"AW00011346","Deanna","Gutierrez","10/13/2008 11:15:00"
```

5. To add or change column names and edit datatypes, click **Edit Columns** and then enter the column name and select the datatype for each column.
6. To specify advanced properties, click **Additional Attributes** and specify the following properties:

Property	Description
Line sequential	Ends each row with a newline character. Must be enabled when you use a fixed-width file format for a flat file source or target object. Line sequential is enabled by default.
Number of rows to skip	Number of initial rows to skip. For example, you might want to skip blank rows or header rows.
Number of bytes to skip after column ending	Number of bytes between the last column of one row and the first column of the next.
Null character type	Whether the null character is text or ASCII.
Null character	Character to represent a null value.
Repeat null character	Reads repeat null characters in a single field as a single null value.

Property	Description
Strip trailing blanks	Removes trailing blanks from string values.
Default Date Format	Date format to use for the column when a date format is not specified in the flat file connection. To specify the default date format, enter the following information: - Whether the field width is adjusted or fixed. Enter A for adjusted width or F for fixed width. - The field width in bytes. - The date format. For example, if you want the format to be fixed width with a width of 12 bytes and date format of DD/MM/YYYY, enter the following text: `F 12 DD/MM/YYYY`

To save the advanced properties, click **OK**.

7. To save the fixed-width file format, click **Save**.

CHAPTER 5

Hierarchical schemas

A hierarchical schema is an asset that is based on a schema file or sample JSON file that you import into Data Integration.

A hierarchical schema is required for the Relational to Hierarchical and Hierarchical to Relational transformations. The schema defines the expected hierarchy of the output data. You can create multiple hierarchical schemas and store them in your project folders.

Note: The hierarchical schema supports XSD schemas with up to 10,000 elements. To process an XSD schema that contains more than 10,000 elements, split the data into two hierarchical schemas.

Creating a hierarchical schema

Create a hierarchical schema in Data Integration.

1. Click **New** > **Components** > **Hierarchical Schema**.
2. In the **New Hierarchical Schema** page, enter a name and description. You must provide a name for the hierarchical schema.
3. Browse to select a project location.
4. To select a schema or sample file, click **Upload**. Click **Choose File** and browse for an XSD file or select a sample JSON file, and then click **OK**.

   When you add a JSON sample file, Data Integration generates a schema from the sample.
5. If you selected an XSD file with multiple possible root elements, select a root from the drop-down menu.
6. If you selected a schema that refers to another schema, you must also upload the referenced schema. To upload the referenced schema, click **Upload**, browse for the referenced schema file and click **OK**.
7. To save the hierarchical schema, click **OK**.

CHAPTER 6

Intelligent structure models

A CLAIRE® intelligent structure model is an asset that is based on a sample file that contains data with little or no structure. Intelligent Structure Discovery, a service that is part of CLAIRE technology, determines the underlying patterns of the sample file and creates a model that can be used to transform, parse, and generate output groups.

Long, complex files with little or no structure can be difficult to parse. Intelligent Structure Discovery can automatically decipher input data and discover the patterns, repetitions, relationships, and types of data in unstructured files.

Intelligent Structure Discovery creates a model that expresses the expected output data. You can use an intelligent structure model in mappings to parse unstructured, semi-structured, or structured data.

After you create an intelligent structure model you can view, edit, refine, and export it. When you save or export an intelligent structure, Intelligent Structure Discovery creates an `.amodel` file.

You can create models from the following input types:

- Delimited files, for example, CSV files
- Machine generated files such as weblogs and clickstreams
- JSON files
- XML files
- ORC files
- Avro files
- Parquet files
- Microsoft Excel files
- Data within PDF form fields
- Data within Microsoft Word tables

Using intelligent structure models in Structure Parser transformations

You can use an intelligent structure model in a Structure Parser transformation in a Data Integration mapping.

You can select the intelligent structure model that the Structure Parser uses, the type of input that the transformation expects to receive, and the output that you pass to downstream transformations. An intelligent structure model is required for Structure Parser transformations. For more information about Structure Parser transformations, see *Transformations*.

Using intelligent structure models in data engineering mappings

You can use an intelligent structure model in a data engineering mapping.

You can add an intelligent structure model to data objects, and then incorporate them into mappings in data engineering. To use an intelligent structure model in a data object, first export it from Data Integration to your local drive.

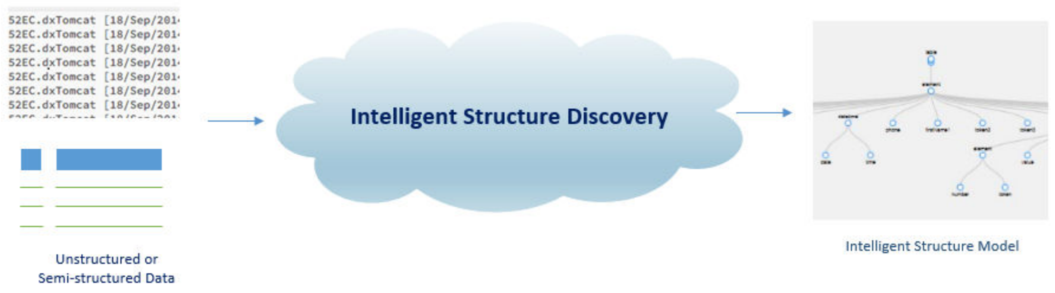
Use Informatica Developer to add the intelligent structure to a complex file data object, Amazon S3 data object, or Microsoft Azure Blob data object. You can add the data object to a data engineering mapping and process data on the Spark engine. For more information, see the *Data Engineering Integration User Guide*.

Intelligent Structure Discovery process

You can create an intelligent structure using Intelligent Structure Discovery.

After you provide a sample file, Intelligent Structure Discovery determines the underlying and repeating patterns of the data and creates a structure that represents the fields of data and their relationships. You can quickly model data for files whose structure is very hard, time consuming, and costly to find, such as log files, clickstreams, customer web access, error text files, or other internet, sensor, or device data that does not follow industry standards.

The following image shows the process by which Intelligent Structure Discovery deciphers the underlying patterns of data and creates a model of the data patterns:


52EC.dxTomcat [18/Sep/201
Unstructured or
Semi-structured Data
Intelligent Structure Discovery
Intelligent Structure Model

After you save an intelligent structure model, you can associate it with a Structure Parser and use in a Data Integration mapping. You can export the intelligent structure model to use it with a data object in a data engineering mapping.

Discovering a file structure and creating an intelligent structure model

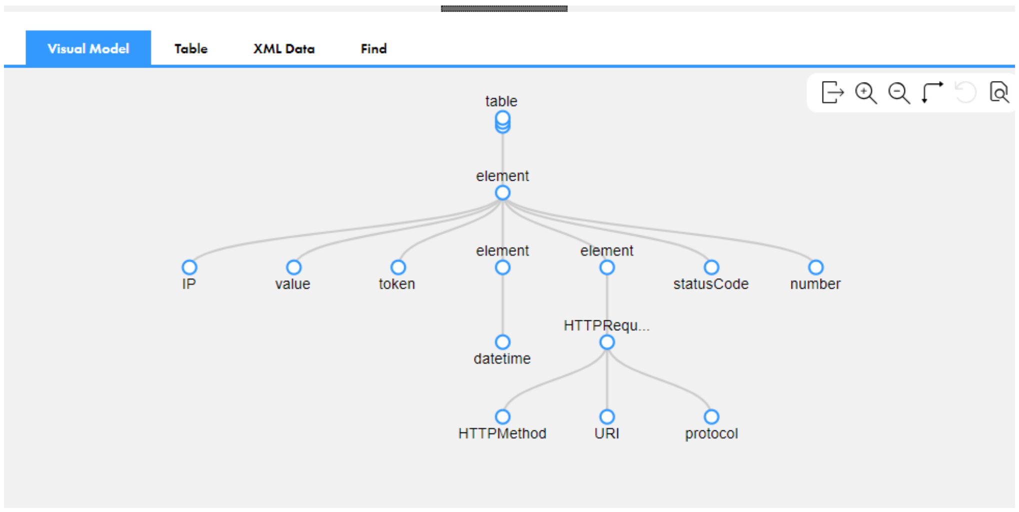
Create an intelligent structure model to parse data in a semi-structured or unstructured file. Use a simplified sample file to generate the model.

1. Click **New** > **Components** > **Intelligent Structure Model**, and then click **Create**.
2. On the **Intelligent Structure Model** page, enter a name for the intelligent structure model.

   The name can contain alphanumeric characters and underscores.
3. Select a location. Navigate to the project and folder where you want to save the intelligent structure model, or use the default folder.

   You can change the name or location after you save the intelligent structure model using the **Explore** page.
4. Browse for a file on which to base the model and click **Discover Structure**.

   Intelligent Structure Discovery deciphers the data in the sample file and discovers the patterns expressed in the data. The following image shows a sample discovered structure on the **Visual Model** tab:

   Intelligent Structure Discovery creates nodes with unique names. If Intelligent Structure Discovery discovers instances of the same type of data, for example a timestamp in two tables, it will assign node names with a number suffix to distinguish between the names, for example **timestamp1** and **timestamp2**.
5. To refine the intelligent structure and customize the nodes in the output data, right-click a node and select an action. The menu includes the actions that are relevant for the node. When you perform an action on a node, the action affects the output group that contains the node.

   For more information, see "Refining a discovered structure" on page 30.
6. To add a prefix or a suffix to a field name or to multiple field names, perform the following actions:
   a. Search for the field name or for a part of the field name in the **Find** pane.

      Fields names that match the search appear in the **Find** pane.
   b. Select the fields to which to add the prefix or suffix, and then, from the **Actions** menu, select **Add Prefix** or **Add Suffix**.

c. Enter the prefix or suffix in the text field to the right of the **Actions** menu, and then click **Apply to Selected**.

   The prefix or suffix can contain alphanumeric characters and underscores.

   Intelligent Structure Discovery adds a prefix or a suffix to the names of the selected fields.

7. For an intelligent structure that is based on a sample JSON file, you can use additional JSON sample files to enrich the structure with new data discovered in the sample files.

   The new nodes are listed in the **Find** pane and highlighted in the model.

8. For an intelligent structure created for an Excel worksheet, metadata nodes with sheet index and name are created and are excluded from the model by default. Select **Include in Structure** to add those nodes to the output.

9. To view the output groups, select the **Table** tab, select the output group, and select an action to perform.

   When you perform an action on a column in an output group, the action applies to the related node in the intelligent structure.

10. Click **Save**.

    Intelligent Structure Discovery generates `.amodel`intelligent structure model, and saves the asset in the selected project and folder.

Selecting a sample file

When you select a file on which to base an intelligent structure, the file should be very similar to files that are used in production, and contain each type of data that you want to parse, in the same format as the data you want to parse.

Use a simplified sample file to generate the model. For example, if the input data has tables, provide a table with just a few sample rows rather than many rows of data. If you use a JSON input file that contains repeating groups of data, limit the number of repetitions.

If the intelligent structure does not match the input file that you plan to use, or only partially matches the input file, there might be a large amount of unidentified data and data loss. However, some variations, such as in date format, will still be parsed.

If your production data varies from the original file used to create the intelligent structure model, the model might still be able to capture the data, depending on the type of variation. For example, you might create a model with a date format such as `18/Sep/2014`, and your data uses a different format, such as `4-Apr-2014`. The model recognizes and parses the data.

The model can also address data drift in certain cases, as in the following example. In this example, the sample data used to create the model contains the following text:

```
96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat [18/Sep/2014:14:54:24 +0300] 'GET /dx-console/
HTTP/1.1' 302 -
96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat [18/Sep/2014:14:54:25 +0300] 'GET /dx-
console/com.informatica.b2b.dx.Main/main.jsp HTTP/1.1' 302 -
96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat [18/Sep/2014:14:54:25 +0300] 'GET /dx-console/
login.jsp HTTP/1.1' 200 4472
```

The data that you parse with the model contains the following text:

```
96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat this_is_new_version_data [4-Apr-2014
05:14:24WIT]             'GET /dx-console/ HTTP/1.1' 302 -
96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat this_is_new_version_data [4-Apr-2014
05:14:24WIT]             'GET /dx-console/com.informatica.b2b.dx.Main/main.jsp HTTP/
1.1' 302 -
```

```
96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat this_is_new_version_data [4-Apr-2014
05:14:24WIT]              'GET /dx-console/login.jsp HTTP/1.1' 200 4472
```

Some of the data has drifted and is in a different location with relationship to the other data. However, the model can parse this variation.

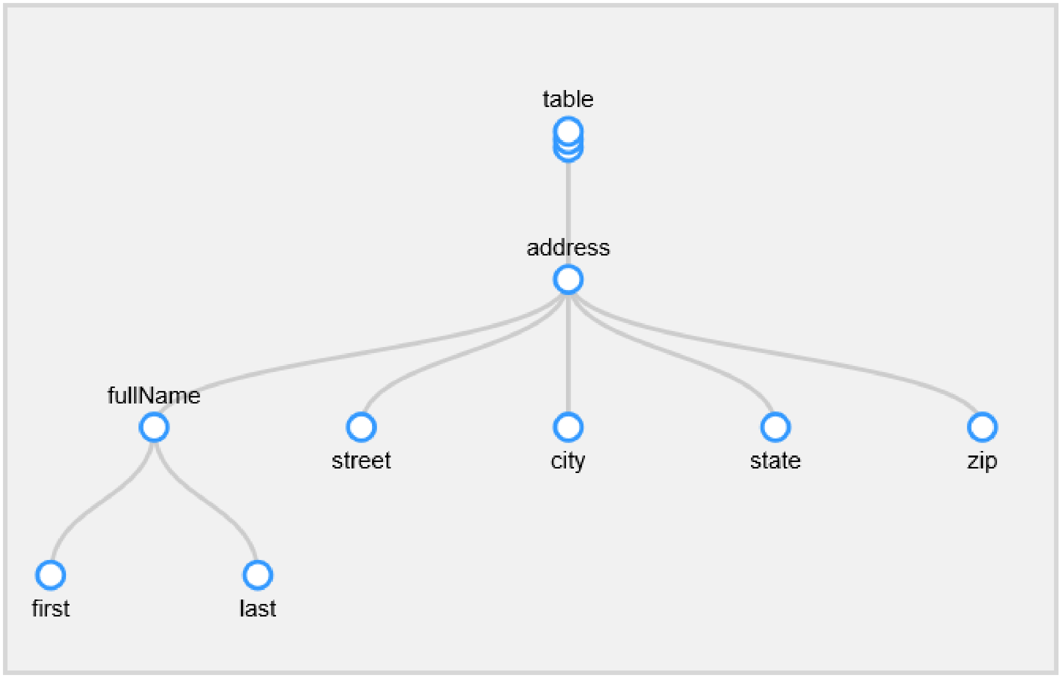
Intelligent structure example

You can create an intelligent structure with a sample file.

As an example, you want to create an intelligent structure for a CSV input file that contains the following content:

```
first,last,street,city,state,zip
Carrine,Stone,17 Torrence Street,Livingston,PA,10173
Poona,Tillkup,52 Perez Avenue,Livingston,PA,10256
Tasha,Herrera,158 Shiraz Boulevard,Kensington,WA,33823
John,Washington,22A Zangville Drive,Tucson,AZ,20198
Jane Hochuli 4483 Central Street Suite 30 Phoenix PA 38721
```

The following image shows a sample discovered structure:



You can see that Intelligent Structure Discovery has created nodes representing the fields in the input file, such as **first**, **last**, **street**, **city**, **state**, and **zip**.

The structure represents not just the data fields themselves, but also the relationship of the fields to each other. For example, Intelligent Structure Discovery recognized that the data `Carrine,Stone` represents the first name and last name of a person. The nodes **first** and **last** are grouped together under the node **fullName**, representing the relationship of the data with each other.

Intelligent Structure Discovery also recognized that the data as a whole represented addresses. The data is grouped under a parent node **address**.

The nodes represent fields that are part of the output. Nodes that are related are grouped into an output group. Output groups can contain one or more nodes.

Refining a discovered structure

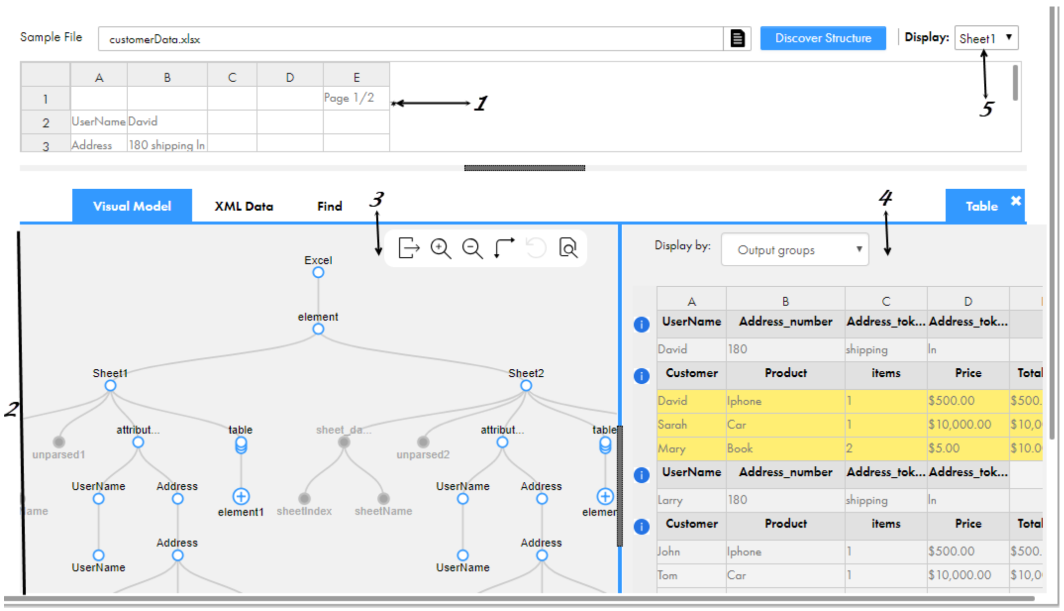
After you discover the file structure, you can view the expected output, refine the nodes and output groups to suit your needs, and then save the model. Use the **Visual Model** tab and the **Table** tab to understand and refine the output.

On the **Visual Model** tab, you can see the output displayed in a graphical, tree-like structure. The intelligent structure shows the discovered types of data as nodes, and shows their relationship to each other in a graphical format. Use the **Visual Model** tab to trace how the input data is mapped to a node, and to perform actions on nodes, such as renaming, combining, or excluding nodes from the output.

On the **Table** tab, you can see the relational output that the intelligent structure produces. The output is organized in one or more output groups. An output group contains one or more nodes. Use the **Table** tab to determine how a node is mapped to an output group. You can also use the **Table** tab to rename nodes or exclude nodes from the output.

Navigating the output

The following image shows part of the **Intelligent Structure Details** page, with the input data panel above, and below, the intelligent structure in the Visual Model tab, and the output groups in the Table tab:



	A	B	C	D	E
1					Page 1/2
2	UserName	David			
3	Address	180 shipping ln			

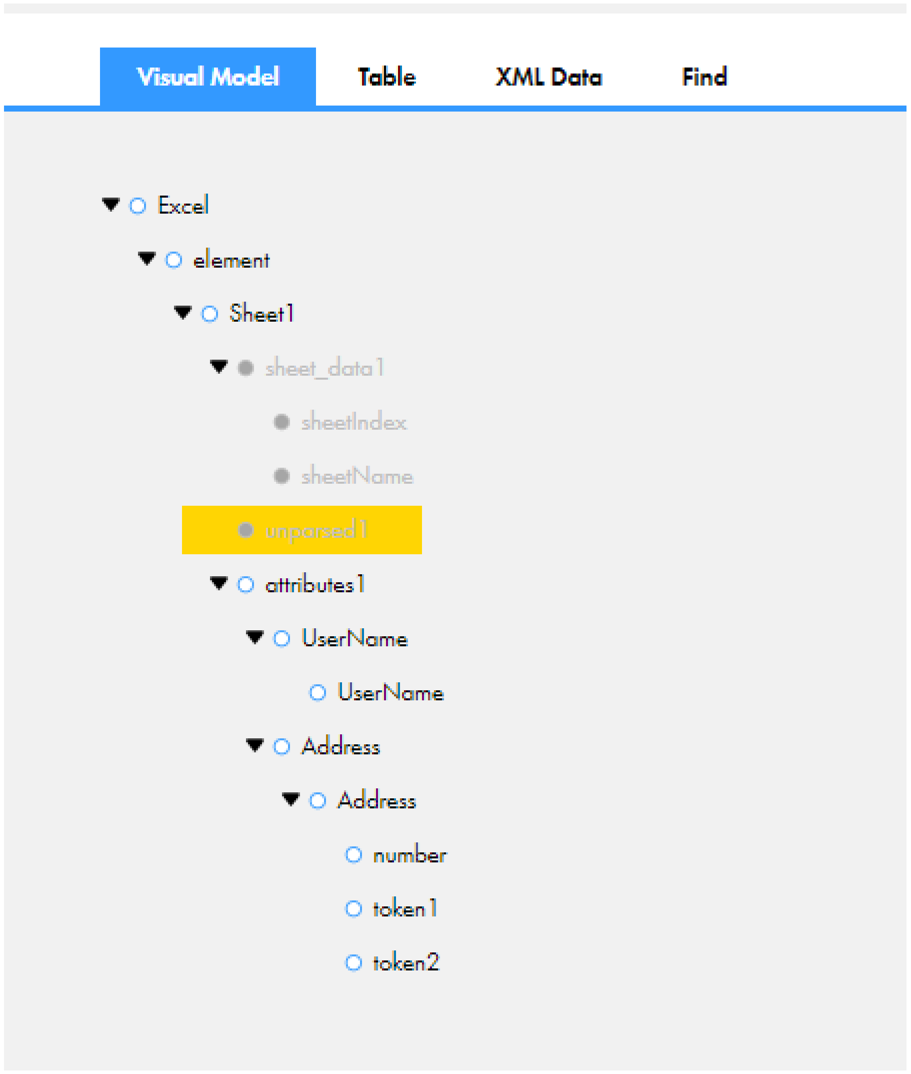
A	B	C	D	
UserName	Address_number	Address_tok...	Address_tok...	
David	180	shipping	ln	
Customer	Product	items	Price	Total
David	Iphone	1	$500.00	$500.
Sarah	Car	1	$10,000.00	$10,0
Mary	Book	2	$5.00	$10.0
UserName	Address_number	Address_tok...	Address_tok...	
Larry	180	shipping	ln	
Customer	Product	items	Price	Total
John	Iphone	1	$500.00	$500.
Tom	Car	1	$10,000.00	$10,0

The following table describes the areas of the Mapping Designer:

Mapping Designer Areas	Description
1. Input data panel	Shows the data from the sample file.
2. Intelligent structure panel	Shows the intelligent structure and output groups.

Mapping Designer Areas	Description
3. Visual Model tab	Shows the intelligent structure, and the following icon menu bar: - Export icon. Export the model. - Zoom In icon. Increases the size of the model. - Zoom Out icon. Decreases the size of model. - Display icon. Toggle model display orientation between landscape and portrait folder view. - Refresh icon. Refresh the model.
4. Table tab	Shows the output groups that the intelligent structure generates.
5. Excel sheet display selection	Select the Excel sheet for which to display data in input data panel . This is displayed only when the sample file is an Excel with multiple sheets.

When you toggle the model display orientation to portrait, the model shows as a folder tree. Each node shows as a level in the tree, and you can select the same node menu options as are available in the model landscape view. The following image shows the model as a folder tree.

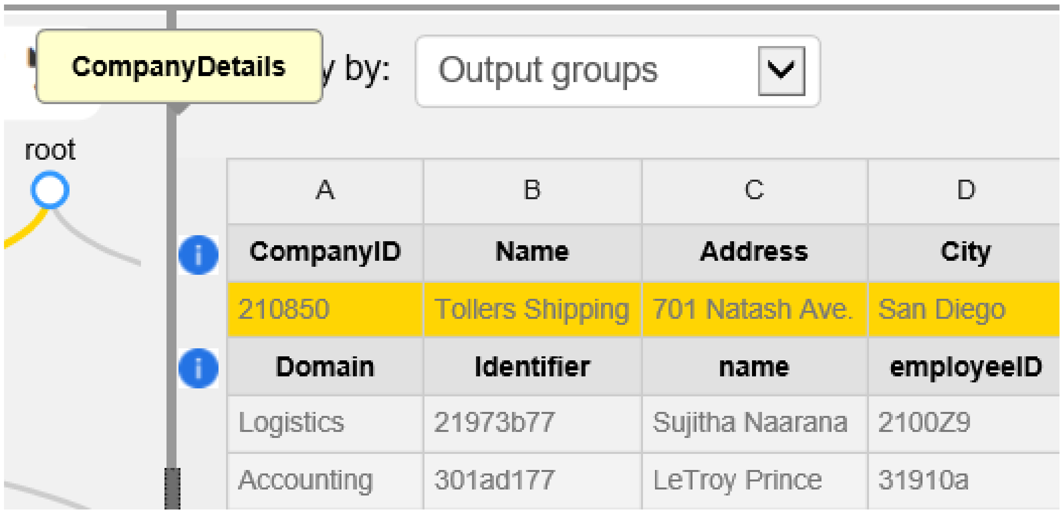


A node in the intelligent structure is represented by a column in an output group. The column contains all the iterations of data that relate to the node.

To better understand the output, you can select a node and view the corresponding data in the input data panel and in the related column in the relevant output group. In this image, two nodes are highlighted, and data can be traced from the input data to the output group. Use the Table tab to determine the type of output the model produces for each node, for groups of nodes, and for the model as a whole.

Each group has an information icon at the left of its column display. When you hover over the information icon, the entire output group is highlighted and the name of the output group is displayed.

The following image shows a highlighted output group named **CompanyDetails** with the name of the output group displayed:



A	B	C	D
CompanyID	**Name**	**Address**	**City**
210850	Tollers Shipping	701 Natash Ave.	San Diego
Domain	**Identifier**	**name**	**employeeID**
Logistics	21973b77	Sujitha Naarana	2100Z9
Accounting	301ad177	LeTroy Prince	31910a

Work with both the Visual Model tab and the Table tab for best results. To change the output, you can adjust and refine the intelligent structure. When you refine the intelligent structure in the Visual Model tab, you can immediately see the effect on the output groups in the Table tab.

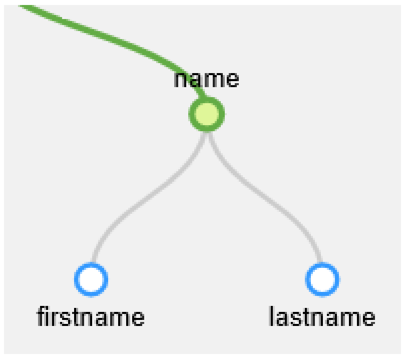
By default, the Table tab shows data according to output groups. You can select to display groups according to incoming records, to see how incoming data is grouped in the intelligent structure. The representation of incoming records might be normalized to address the presence of repeating elements.

For models based on an Excel file that contains multiple sheets, all sheets are displayed in the visual model. You can select which sheet to display in the input data panel.

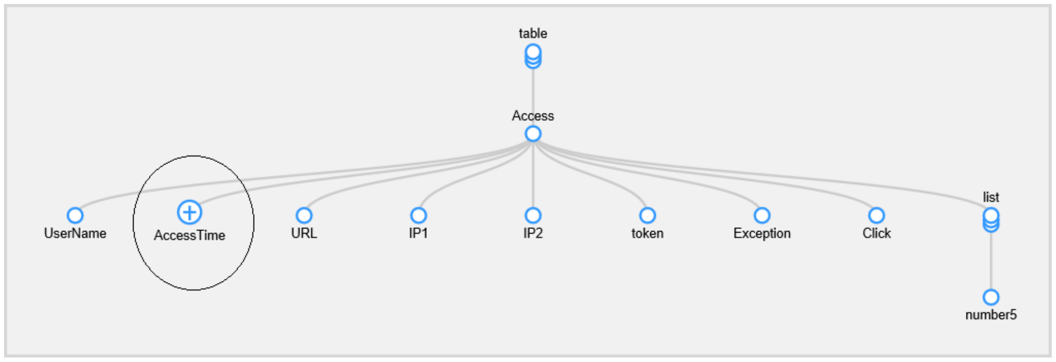
Navigating the intelligent structure

You can enlarge or shrink the intelligent structure in the Visual Model tab to better view the nodes. You can drag the model across the screen to focus on a specific part of the model.

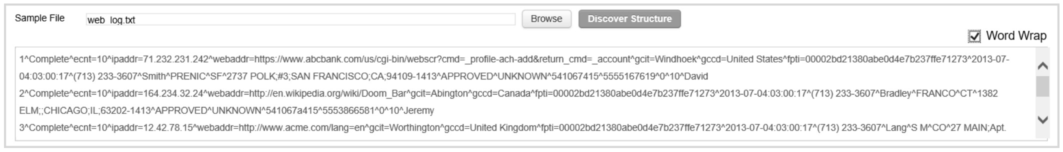
A grouping of data fields is displayed as a node with child nodes. For example, the input data might contain the text `Tatanya Morales`, a first and last name. The intelligent structure represents the name with a `name` parent node, a `firstname` child node, and a `lastname` child node. The following image shows the parent and child nodes:



You can collapse parent nodes, for example, if an intelligent structure is large, and contains parent nodes with very many child nodes. When you collapse nodes, all child nodes remain in the intelligent structure but are not displayed. To collapse a node, click the node and select **Collapse**. The following image shows a model with a collapsed node:

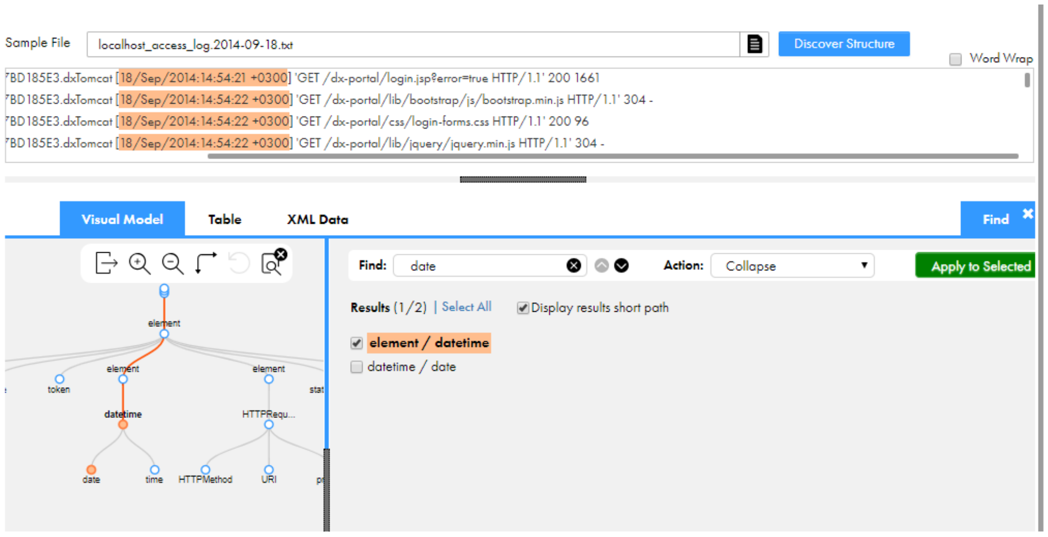


You might find it difficult to trace the input for the intelligent structure if the input has very lengthy rows of data. You can select to view the input in word-wrap format. If you do not word wrap lengthy input rows, only part of each row is displayed. Wrapping the text does not affect the intelligent structure or the input format. To wrap text, select **Word wrap**. The following image shows data in word wrap mode:



```
1^Complete^ecnt=10^ipaddr=71.232.231.242^webaddr=https://www.abcbank.com/us/cgi-bin/webscr?cmd=_profile-ach-add&return_cmd=_account^gcit=Windhoek^gccd=United States^fpti=00002bd21380abe0d4e7b237ffe71273^2013-07-
04:03:00:17^(713) 233-3607^Smith^PRENIC^SF^2737 POLK;#3;SAN FRANCISCO;CA;94109-1413^APPROVED^UNKNOWN^541067415^5555167619^0^10^David
2^Complete^ecnt=10^ipaddr=164.234.32.24^webaddr=http://en.wikipedia.org/wiki/Doom_Bar^gcit=Abington^gccd=Canada^fpti=00002bd21380abe0d4e7b237ffe71273^2013-07-04:03:00:17^(713) 233-3607^Bradley^FRANCO^CT^1382
ELM;;CHICAGO;IL;63202-1413^APPROVED^UNKNOWN^541067a415^5553866581^0^10^Jeremy
3^Complete^ecnt=10^ipaddr=12.42.78.15^webaddr=http://www.acme.com/lang=en^gcit=Worthington^gccd=United Kingdom^fpti=00002bd21380abe0d4e7b237ffe71273^2013-07-04:03:00:17^(713) 233-3607^Lang^S M^CO^27 MAIN;Apt.
```

You can locate specific nodes in large or complex intelligent structures. Click **Find** to open the Find pane and enter the node name string you want to locate. A list of nodes whose name contains the string is displayed and those nodes are highlighted in the model. The list shows the short path with the last 2 nodes of the path. To show the full path to the nodes unselect **Display results short path**.



localhost_access_log.2014-09-18.txt

```
7BD185E3.dxTomcat [18/Sep/2014:14:54:21 +0300] 'GET /dx-portal/login.jsp?error=true HTTP/1.1' 200 1661
7BD185E3.dxTomcat [18/Sep/2014:14:54:22 +0300] 'GET /dx-portal/lib/bootstrap/js/bootstrap.min.js HTTP/1.1' 304 -
7BD185E3.dxTomcat [18/Sep/2014:14:54:22 +0300] 'GET /dx-portal/css/login-forms.css HTTP/1.1' 200 96
7BD185E3.dxTomcat [18/Sep/2014:14:54:22 +0300] 'GET /dx-portal/lib/jquery/jquery.min.js HTTP/1.1' 304 -
```

Find: date
Action: Collapse

Results (1/2) | Select All
[✓] Display results short path

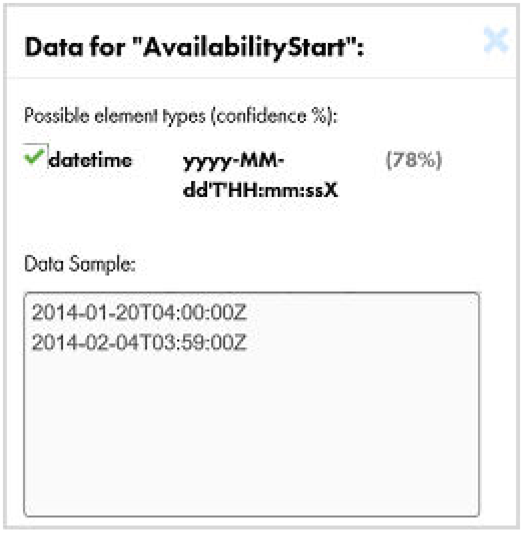
[✓] element / datetime
[ ] datetime / date

After you locate specific nodes in a structure you can then select from the list the nodes of interest and select one of the actions on those nodes:

- Exclude. Excludes the nodes from the visual model
- Include. Includes the nodes in the visual model
- Expand. Shows the nodes that had been collapsed
- Collapse. Hides the portion of the model under the nodes
- Change Type to String. Changes the type of the node to string
- Replace. Replaces the string in the node names with a string that you enter

Click **Apply to Selected** to perform the action on the nodes you selected.

Beside tracing the node input data, you might want to find out other information about a node. You can view the accuracy of the algorithm results, possible node types related to the node, and sample data in the input file that corresponds to the node. To view information about a node, click the node and select **Open Data**. The following image shows data for a node named `AvailabilityStart`:



Data for "AvailabilityStart":

Possible element types (confidence %):

✓ datetime	yyyy-MM-dd'THH:mm:ssX	(78%)

Data Sample:

```
2014-01-20T04:00:00Z
2014-02-04T03:59:00Z
```

Editing the nodes

You can change the output by editing the intelligent structure nodes.

Perform the following actions to edit nodes on the Visual Model tab:

Rename a node.

To rename a node, right-click a node, select **Rename**, and then type in a name, for example **Street Address**. Node names are not case sensitive.

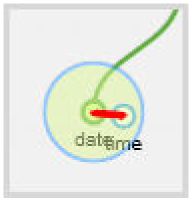
Note: If you try to rename a node to the name of an existing node, Intelligent Structure Discovery does not accept the name change.

Apply unique names to nodes

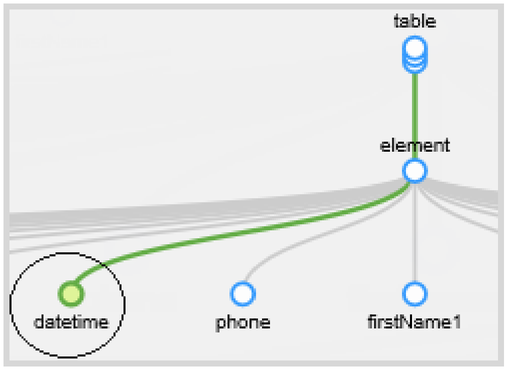
To apply unique naming across the structure, right-click the root node and select **Apply Unique Naming**. When you apply unique naming to a model, if the data in the sample files contains fields with identical names in different groups,Intelligent Structure Discovery uses suffixes to rename identical names to unique names.

Combine data in two nodes.

To join two nodes, click and drag a node to the other node. For example, in the following model, the **time** node is selected and dragged to the **date** node:



The following image shows the combined node:



Combine all child node data into a parent node.

To flatten a parent node and merge its child nodes, right-click the node and select **Flatten**. When you flatten a node, the data from the child nodes is merged into one output field for the parent node. The child nodes are no longer separate fields in the output.

Exclude a node from the output.

To exclude a node from the output, right-click the node and select **Exclude from Structure**. When you exclude a node from the model, the data from the node is not part of the output of the model, and is not parsed during run time. To re-include a node that you excluded, right-click the node and select **Include in Structure**.

Change the output data type.

To change the output data type for a node, perform the following steps:

1. Right-click the node and select **Open data**. Alternatively, double-click the node.
2. In the **Element Type** display field, select the data type from the drop-down list of available types and then click **OK**.

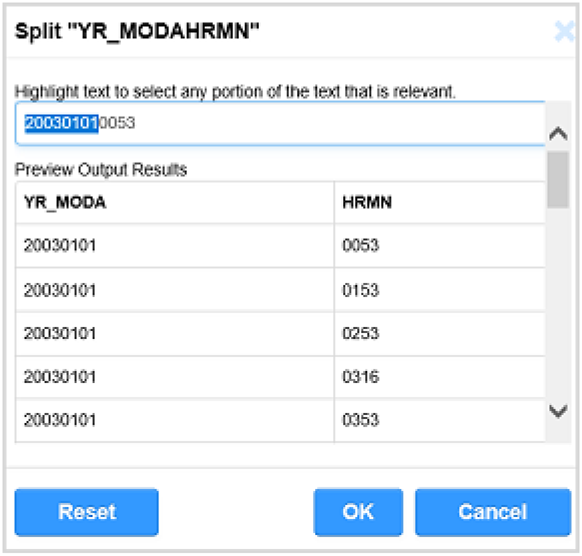
The data type selection applies to an intelligent structure model that you export to Informatica Developer.

Split a node.

To split the data in a node into two nodes, for example, for a credit card number for which you only want to store the last four digits, perform the following steps:

1. Right-click the node and select **Split**. The **Split** dialog box appears.
2. In the **Split** dialog box, highlight the data in the display field to indicate where to split the data. The data is split and displayed as two fields, with one field containing the highlighted data, and the second field containing the non-highlighted data.
3. To confirm the split, click **OK**.
4. To undo the split, click **Reset**.

For example, you want to split a node with year-month-date-hour-minute data in one field into two fields. You want to create a year-month-date field and an hour-minute field. The following image shows the highlighted section that you want to split:


Split "YR_MODAHRMN"

Highlight text to select any portion of the text that is relevant.

200301010053

Preview Output Results

YR_MODA	HRMN
20030101	0053
20030101	0153
20030101	0253
20030101	0316
20030101	0353

Treat JSON model as repeating

By default, when the sample JSON model does not contain repetitions, the Structure Parser transformation does not parse repeating groups. If you want to parse repeating groups, right-click the JSON root node and select **Treat as Recurring**.

Treat an element as repeating

By default, when an element in the sample JSON or XML model does not contain repetitions, the Structure Parser transformation does not parse repetitions of the element. If you want to parse repetitions of the element, right-click the node and select **Treat as a List**. To undo a **Treat as a List** action, right-click the node and select **Treat as Single**.

Enriching an existing intelligent structure with new samples

After you create an intelligent structure from one sample file, you can use additional sample files to enrich the structure with new fields that exist in the new samples.

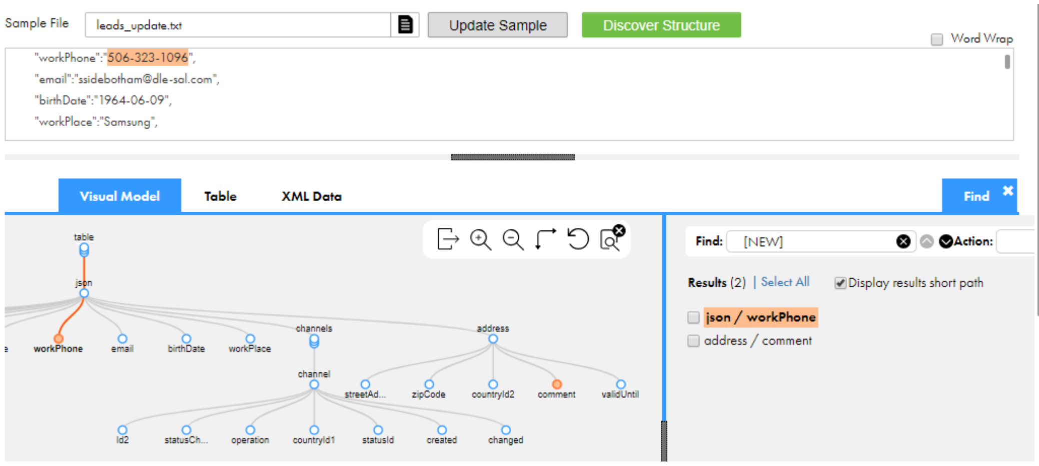
For example, if the data you want to map contains data that is not in the structure and therefore is not mapped, you can add that data to the structure based on a new sample file that contains that data.

You can enrich existing structures that are based on a JSON, XML, ORC, AVRO, and PARQUET sample files. The files that you use to enrich the structure must be of the same file type as the type of file that the structure is based on. For example, to enrich a structure that is based on a JSON sample file use additional JSON sample files.

To add data to the structure click **Update Sample** and select the new sample file.

Intelligent Structure Discovery creates nodes for new data in the sample file. To facilitate locating the new nodes, the **Find** pane opens with the search pattern **[NEW]**. The list of new nodes shows and those nodes are highlighted in the structure.

Note: If the new sample does not contain all the data that was in the structure, no data is removed from the structure and the entire structure is displayed. However, the input data panel and the **Table** pane only show data that is in the new sample.



```
"workPhone":"506-323-1096",
"email":"ssidebotham@dle-sal.com",
"birthDate":"1964-06-09",
"workPlace":"Samsung",
```

Find: [NEW]

Results (2) | Select All [✓] Display results short path

- json / workPhone
- address / comment

When you save the model the structure is updated with the new data.

Document identifiers

You can add document identifiers to data in a model to identify the sample file that the data is based on. Each sample file is identified by a unique ID. Intelligent Structure Discovery adds document identifiers to groups in the model.

For example, add document identifiers to a model if you want the Structure Parser transformation to join output groups that are based on the same file and output them to a single target.

To add document identifiers to the intelligent structure, select the top node in an intelligent structure, right-click the node, and select **Add a Document ID**. To remove document identifiers, right-click the node and select **Remove a Document ID**.

In the model, document identifier fields appear in the Table tab, in the following format: `<group name>_documentId`. For example, for a group named "organization" the document identifier field is `organization_documentId`.

In the Structure Parser transformation, a document identifier output port appears for each output group.

Working with output groups

You can use the **Table** tab to view, understand, and edit the expected output groups for the intelligent structure model. You can also use the **Table** tab to view the input records that Intelligent Structure Discovery processes.

You can select the following display options on the Table tab:

Display output groups.

To view the way data is grouped together in the output, select to view output groups.

Display incoming records.

To understand how Intelligent Structure Discovery receives input records, select to view incoming records. The display might include normalization of groups to address the presence of repeating elements. This does not reflect the grouping of elements in the output groups.

You can perform the following actions on output groups:

Rename a node.

To rename a node, right-click the column that represents the node, and type a name, for example `City`.

Exclude a node.

To exclude a node from an output group, right-click the column and select **Exclude from Structure**. If you exclude the node, the data is not parsed in run time.

Note: The **Table** tab can display up to 200 rows of data. If you have a large amount of data or many repeating groups, the **Table** tab might not display all the output groups.

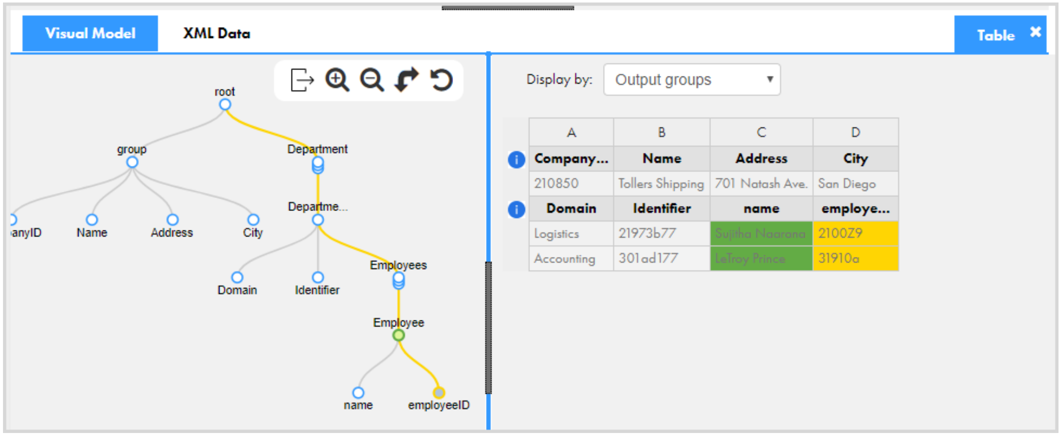
Repeating groups

Intelligent Structure Discovery creates a repeating group for input data such as a table or array, where a row or set of fields repeats as a group.

For example, you create a model for the following JSON input:

```
{
  "CompanyID": 210850,
  "Name": "Tollers Shipping",
  "Address": "701 Natash Ave.",
  "City": "San Diego",
  "Department": [
    {
      "Domain": "Logistics",
      "Identifier": "21973b77",
      "Employees": [
        {
          "name": "Sujitha Naarana",
          "employeeID": "2100Z9"
        }
      ],
    },
            {
      "Domain": "Accounting",
      "Identifier": "301ad177",
      "Employees": [
        {
          "name": "LeTroy Prince",
          "employeeID": "31910a"
        }
      ]
    }
  ]
}
```

The following image shows the model you created with the JSON input:



Display by: Output groups

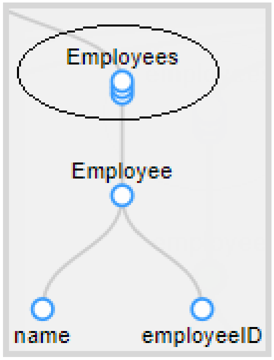
A	B	C	D
Company...	**Name**	**Address**	**City**
210850	Tollers Shipping	701 Natash Ave.	San Diego
Domain	**Identifier**	**name**	**employe...**
Logistics	21973b77	Sujitha Naarana	2100Z9
Accounting	301ad177	LeTroy Prince	31910a

The model contains an **Employees** repeating node that contains an **Employee** group node. The **Employee** group node has a **name** child node and an **employeeID** child node.

If you examine the image, in the Visual tab, the **employeeID** child node in the **Employees** repeating group is highlighted. In the Table tab, the **DepartmentItem** output group contains the **employeeID** node. The columns that relate to the **employeeID** node are highlighted.

A repeating group is displayed in the Visual Model with three overlapping circles for the parent node.

The following image shows the **Employees** repeating group:

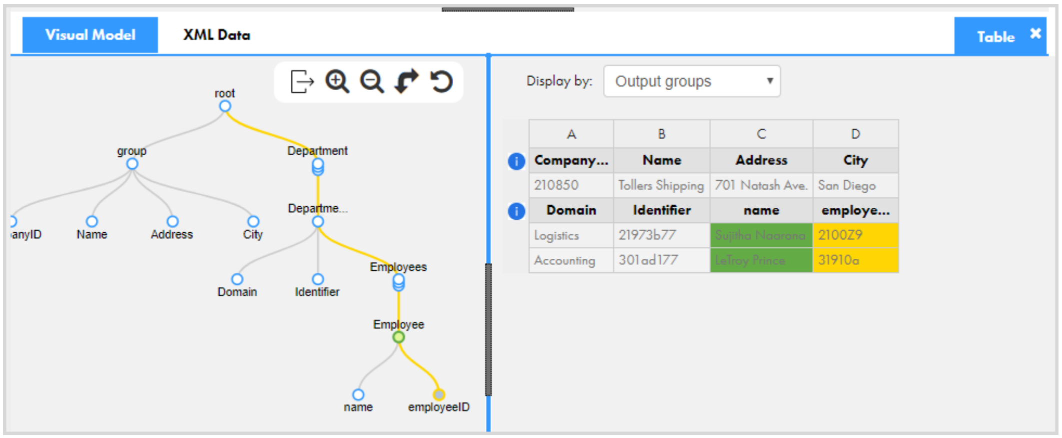


Primary and foreign keys

For models that contain a repeating group within a repeating group, an output group can include a column called the primary key. A foreign key is a column in another output group that points to the primary key of the first output group. You identify the relationship between a row in one output group and a row in another output group using the keys.

Before Intelligent Structure Discovery adds keys, you must assign a nested repeating node to its own output group. Intelligent Structure Discovery adds a primary key to the parent and a foreign key to the child group.

The following image shows the model created with the JSON input. The **Employees** repeating group is highlighted:



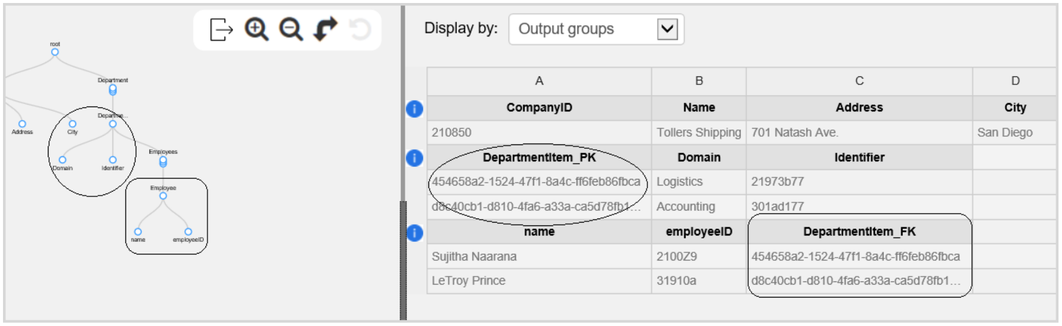
Display by: Output groups

A	B	C	D
Company...	**Name**	**Address**	**City**
210850	Tollers Shipping	701 Natash Ave.	San Diego
Domain	**Identifier**	**name**	**employe...**
Logistics	21973b77	Sujitha Naarana	2100Z9
Accounting	301ad177	LeTroy Prince	31910a

In this model, the **Employee** group is part of the **DepartmentItem** output group, the second group displayed in the Table tab.

You can promote a nested group in the Visual Model tab. You can view the primary and foreign keys in the Table tab.

The following image shows the output groups after you promote the **Employee** group:



Display by: Output groups

A	B	C	D
CompanyID	**Name**	**Address**	**City**
210850	Tollers Shipping	701 Natash Ave.	San Diego
DepartmentItem_PK	**Domain**	**Identifier**	
454658a2-1524-47f1-8a4c-ff6feb86fbca	Logistics	21973b77	
d8c40cb1-d810-4fa6-a33a-ca5d78fb1...	Accounting	301ad177	
name	**employeeID**	**DepartmentItem_FK**	
Sujitha Naarana	2100Z9	454658a2-1524-47f1-8a4c-ff6feb86fbca	
LeTroy Prince	31910a	d8c40cb1-d810-4fa6-a33a-ca5d78fb1...	

Intelligent Structure Discovery generates two separate output groups, the **DepartmentItem** output group, and the **Employee** output group.

Intelligent Structure Discovery added a primary key named **DepartmentItem_PK** to the **DepartmentItem** output group, and a foreign key named **DepartmentItem_PK** to the **Employee** output group.

You can select a different node as the primary key by defining it as a record ID. When you change the record ID, Intelligent Structure Discovery creates a corresponding foreign key in the nested group.

Working with repeating groups

The output for groups is relational tables. When the intelligent structure contains repeating groups that relate to each other, you can configure the relational output to contain different tables for nested groups, as well as select a record ID .

When an intelligent structure contains a repeating group within a repeating group, you can perform the following actions on the Visual Model tab:

Promote to group.

Assign a nested repeating group to its own output group. When you promote a group nested within a repeating group, Intelligent Structure Discovery creates a record ID in the parent output group and a foreign key in the child output group. To promote a group to an independent output group, right-click the nested repeating node and select **Promote to Group**.

Join to parent group.

Re-include a nested group in the parent group after you assigned it to its own output group. To re-include a group, right-click the nested repeating node and select **Join to Parent Group**.

Select as record ID.

Change the record ID. Select a node in the parent group and assign it as a record ID. Intelligent Structure Discovery automatically changes the foreign key in the nested group. To assign a node as record ID, right-click the node and select **Select as Record ID**.

Deselect as record ID.

Reinstate the initial record ID and foreign key for the repeating groups. To deselect a node as the record ID, right-click the node and select **Deselect as Record ID**.

Editing an Intelligent Structure Model

After you save an intelligent structure model, you can edit the basic details of the model.

1. On the **Explore** page, navigate to the project and folder with the intelligent structure model.

   The **Explore** page displays all the assets in the folder.
2. Click to select the row that contains the relevant intelligent structure model. In the **Actions** menu, select **Edit**. In the **Intelligent Structure Details** panel, click **Edit**.
3. You can change the name, description, and location for the model. You must provide a name, as well as a project and folder location, for the model.
4. To save the intelligent structure, click **Save**.

Exporting an intelligent structure model

Before you can use an intelligent structure model in a data engineering mapping, you must export it. You export the model to your local drive, and can then incorporate the model into data engineering data objects.

To export an intelligent structure model, you must have the appropriate license.

1. On the **Explore** page, navigate to the project and folder with the intelligent structure model.

   The **Explore** page displays all the assets in the folder.
2. Click to select the row that contains the relevant intelligent structure model. In the **Actions** menu, select **Edit**. In the **Intelligent Structure Details** panel, click **Edit**.

   The intelligent structure model appears in a separate page.
3. In the **Intelligent Structure** page, find the icon menu in the upper right corner of the Visual Model tab, and click the **Export model** icon.

   Intelligent Structure Discovery downloads the `.amodel` file to your local drive.

Deleting an Intelligent Structure Model

Before you delete an intelligent structure model, verify that no one in the organization needs it.

1. On the **Explore** page, navigate to the project and folder with the intelligent structure model.

   The **Explore** page displays all the assets in the folder.
2. Click to select the row that contains the relevant intelligent structure model. In the **Actions** menu, select **Delete**.

   Note: If an intelligent structure model is used in a mapping, remove it from the mapping before you delete it.

Troubleshooting intelligent structure models

Consider the following troubleshooting tips when you create intelligent structure models.

Using differently structured files causes data loss.

If the intelligent structure model does not match the input file or only partially matches the input file, there might be data loss.

For example, you created a model for a sample file that contains rows with six fields of data, `computer ID`, `computer IP address`, `access URL`, `username`, `password`, and `access timestamp`. However, some of the input files contained rows with eight fields of data, that is a `computer ID`, `computer name`, `computer IP address`, `country of origin`, `access URL`, `username`, `password`, `access code`, and `access timestamp`. The data might be misidentified and some data might be designated as unidentified data.

If some input files contain more types of data than other input files, or different types of data, for best results create a sample file that contains all the different types of data.

Data from PDF forms was not modeled or parsed.

An intelligent structure model can model and parse the data within PDF form fields but not data outside the fields. A field title, or other data outside the field, will not be identified.

Data from Microsoft Word was not modeled or parsed.

An intelligent structure model can model and parse data within Microsoft Word tables. All other data is collected as unparsed data.

Error: Unsupported field names might cause data loss.

Do not use duplicate names for different elements.

If you use Big Data Management 10.2.1, ensure that the names of output groups follow Informatica Developer naming conventions. An element name must contain only English letters (A- Z, a-z), numerals (0-9), and underscores. Do not use reserved logical terms, and do not start element names with a number.

In later versions of Big Data Management or Data Engineering Integration, Intelligent Structure Discovery replaces special characters in element names with underscores and inserts underscores before element names that start with numerals and before element names that are reserved logical terms.

Use Case

You work in an operations group for a manufacturing company. Your team wants to process web logs from your server farms to obtain operations analytics and to identify maintenance issues.

Your back-end system collects data regarding server access and system load in your server farms. Your team wants to identify the operations that have created the most server load in the past few weeks. You want to store data afterwards for auditing purposes.

Before your data analysts can begin working with the data, you need to parse the data. However, the logs are semi-structured, and after server upgrades the log file structure might change slightly and some of the information might take a different format. With a standard transformation, this would cause data loss or log processing failures.

Your initial log files have the following structure:

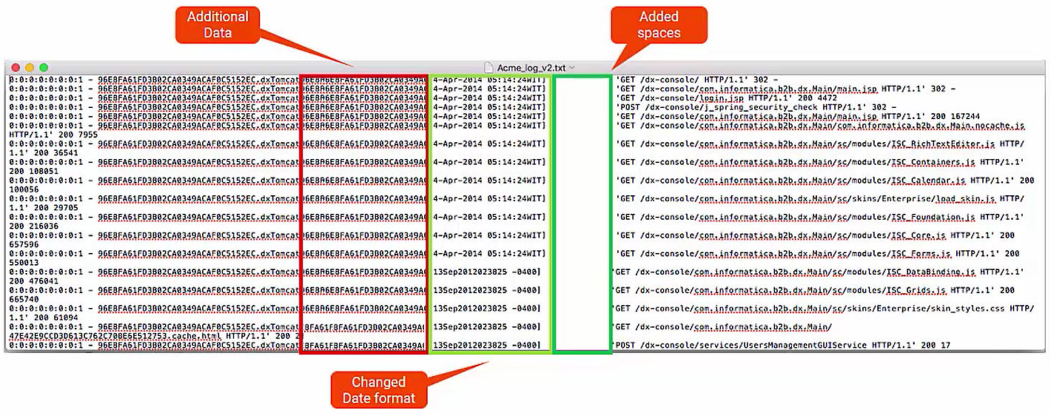
```
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat [18/Sep/2014:14:54:24 +0300]
'GET /dx-console/ HTTP/1.1' 302 -
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat [18/Sep/2014:14:54:25 +0300]
'GET /dx-console/com.informatica.b2b.dx.Main/main.jsp HTTP/1.1' 302 -
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat [18/Sep/2014:14:54:25 +0300]
'GET /dx-console/login.jsp HTTP/1.1' 200 4472
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat [18/Sep/2014:14:55:47 +0300]
'POST /dx-console/j_spring_security_check HTTP/1.1' 302 -
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat [18/Sep/2014:14:55:47 +0300]
'GET /dx-console/com.informatica.b2b.dx.Main/main.jsp HTTP/1.1' 200 167244
```

Following server upgrades, some log files have the following structure:

```
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 96E8FA61FD3B02CA0349 [4-
Apr-2014 05:14:24WIT]              'GET /dx-console/ HTTP/1.1' 302 -
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 96E8FA61FD3B02CA0349 [4-
Apr-2014 05:14:24WIT]              'GET /dx-console/com.informatica.b2b.dx.Main/main.jsp
HTTP/1.1' 302 -
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 96E8FA61FD3B02CA0349 [4-
Apr-2014 05:14:24WIT]              'GET /dx-console/login.jsp HTTP/1.1' 200 4472
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 96E8FA61FD3B02CA0349 [4-
Apr-2014 05:14:24WIT]              'POST /dx-console/j_spring_security_check HTTP/1.1'
302 -
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 96E8FA61FD3B02CA0349 [4-
Apr-2014 05:14:24WIT]              'GET /dx-console/com.informatica.b2b.dx.Main/main.jsp
HTTP/1.1' 200 167244
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 96E8FA61FD3B02CA0349 [4-
Apr-2014 05:14:24WIT]              'GET /dx-console/com.informatica.b2b.dx.Main/
com.informatica.b2b.dx.Main.nocache.js HTTP/1.1' 200 7955
0
```

The data format varies, and some of the data has drifted to a different location.

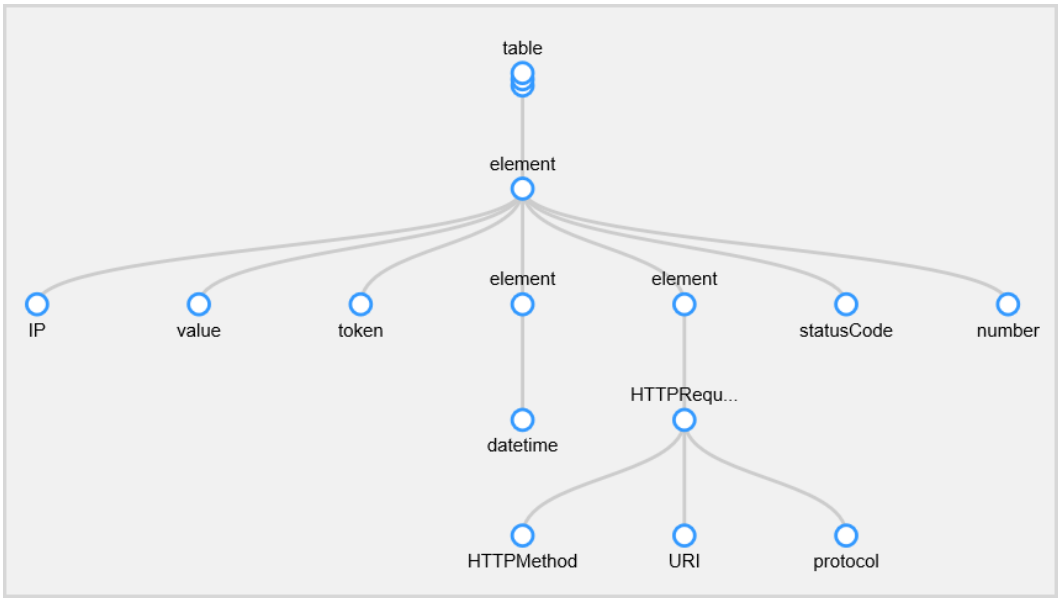
The following image shows the data variations:



```
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'GET /dx-console/ HTTP/1.1' 302 -
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'GET /dx-console/com.informatica.b2b.dx.Main/main.jsp HTTP/1.1' 302 -
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'GET /dx-console/login.jsp HTTP/1.1' 200 4472
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'POST /dx-console/j_spring_security_check HTTP/1.1' 302 -
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'GET /dx-console/com.informatica.b2b.dx.Main/main.jsp HTTP/1.1' 200 167244
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'GET /dx-console/com.informatica.b2b.dx.Main/com.informatica.b2b.dx.Main.nocache.js
HTTP/1.1' 200 7955
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'GET /dx-console/com.informatica.b2b.dx.Main/sc/modules/ISC_RichTextEditor.js HTTP/
1.1' 200 36541
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'GET /dx-console/com.informatica.b2b.dx.Main/sc/modules/ISC_Containers.js HTTP/1.1'
200 108051
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'GET /dx-console/com.informatica.b2b.dx.Main/sc/modules/ISC_Calendar.js HTTP/1.1' 200
100056
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'GET /dx-console/com.informatica.b2b.dx.Main/sc/skins/Enterprise/load_skin.js HTTP/
1.1' 200 29705
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'GET /dx-console/com.informatica.b2b.dx.Main/sc/modules/ISC_Foundation.js HTTP/1.1'
200 216036
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'GET /dx-console/com.informatica.b2b.dx.Main/sc/modules/ISC_Core.js HTTP/1.1' 200
657596
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 4-Apr-2014 05:14:24WIT]         'GET /dx-console/com.informatica.b2b.dx.Main/sc/modules/ISC_Forms.js HTTP/1.1' 200
550013
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 13Sep201023825 -0400]         'GET /dx-console/com.informatica.b2b.dx.Main/sc/modules/ISC_DataBinding.js HTTP/1.1'
200 476041
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 13Sep201023825 -0400]         'GET /dx-console/com.informatica.b2b.dx.Main/sc/modules/ISC_Grids.js HTTP/1.1' 200
665740
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 6E8A6E8FA61FD3B02CA0349A 13Sep201023825 -0400]         'GET /dx-console/com.informatica.b2b.dx.Main/sc/skins/Enterprise/skin_styles.css HTTP/
1.1' 200 61094
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 8FA61F8FA61FD3B02CA0349A 13Sep201023825 -0400]         'GET /dx-console/com.informatica.b2b.dx.Main/
47E42E9CCD3D613C762C70BE6E512753.cache.html HTTP/1.1' 200 2
0:0:0:0:0:0:0:1 - 96E8FA61FD3B02CA0349ACAF0C5152EC.dxTomcat 8FA61F8FA61FD3B02CA0349A 13Sep201023825 -0400]         'POST /dx-console/services/UsersManagementGUIService HTTP/1.1' 200 17
```

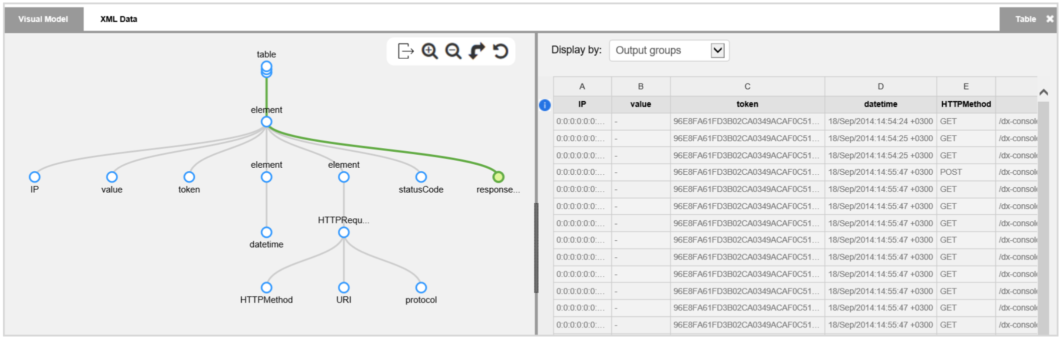
Instead of manually creating individual transformations, your team can generate an intelligent structure model to determine the relevant data sets. You create an intelligent structure in Intelligent Structure Discovery and automatically identify the structure of the data.

The following image shows the intelligent structure that you create:



When you examine the data, you realize that the final element in the model, `number`, actually represents the server response size, or system load. You change the element name to `responseSize`.

The following image shows the updated intelligent structure:



Display by: Output groups

A	B	C	D	E	
IP	value	token	datetime	HTTPMethod	
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:54:24 +0300	GET	/dx-consol
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:54:25 +0300	GET	/dx-consol
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:54:25 +0300	GET	/dx-consol
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:55:47 +0300	POST	/dx-consol
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:55:47 +0300	GET	/dx-consol
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:55:47 +0300	GET	/dx-consol
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:55:47 +0300	GET	/dx-consol
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:55:47 +0300	GET	/dx-consol
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:55:47 +0300	GET	/dx-consol
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:55:47 +0300	GET	/dx-consol
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:55:47 +0300	GET	/dx-consol
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:55:47 +0300	GET	/dx-consol
0:0:0:0:0:0:...	-	96E8FA61FD3B02CA0349ACAF0C51...	18/Sep/2014:14:55:47 +0300	GET	/dx-consol

After you save the intelligent structure as an intelligent structure model, you create a Structure Parser transformation and assign the model to it. You can add the transformation to a Data Integration mapping with a source, target, and other transformations. After the mapping fetches data from a source connection, such as Amazon S3 input buckets, the Structure Parser processes the data with an intelligent structure model. The transformation passes the web log data to downstream transformations for further processing, and then to a target, such as Amazon S3 output buckets.

CHAPTER 7

Mapplets

A mapplet is reusable transformation logic that you can use to transform source data before it is loaded into the target. If you are an Informatica PowerCenter user, you can import a PowerCenter mapplet to use in Data Integration.

You can create a mapplet in one of the following ways:

- Create a mapplet in Data Integration.
- Import a mapplet from PowerCenter. For information about importing a mapplet, see "PowerCenter mapplets" on page 48.
- Generate an SAP BAPI or IDoc mapplet. For information about configuring and importing SAP mapplets, see the help for SAP Connector.

Create mapplets in Data Integration in the Mapplet Designer in the same way that you create mappings in the Mapping Designer. After you create a mapplet, you can add it to a Mapplet transformation to use its transformation logic.

You can use a mapplet in another mapplet. However, you cannot cyclically reference a mapplet. For example, if Mapplet A uses Mapplet B, Mapplet B cannot also use Mapplet A.

To create and use mapplets, your organization must have the appropriate license.

Active and passive mapplets

Mapplets can be either active or passive.

Passive mapplets contain a single input group, a single output group, and only passive transformations.

Active mapplets contain at least one active transformation.

When you use a mapplet in a Mapplet transformation, the mapplet type displays on the **Mapplet** tab.

Mapplet Input and Output

To use a mapplet in a Mapplet transformation, you must configure the mapplet input and output.

A mapplet receives input from an upstream transformation through an Input transformation, from a Source transformation, or both.

A mapplet passes output to a downstream transformation through an Output transformation, to a Target transformation, or both.

A mapplet must contain at least one Input transformation or one Output transformation.

Mapplet input

Mapplet input can be an Input transformation, a Source transformation, or both.

Use an Input transformation when you want the mapplet to receive input data from one or more upstream transformations. You can use multiple Input transformations in a mapplet. When you use the mapplet in a Mapplet transformation, each Input transformation becomes an input group. Use multiple Input transformations when you have multiple pipelines in a mapplet, or when you want the mapplet to receive input from multiple upstream transformations.

You include one or more Source transformations in a mapplet to provide source data. When you use only Source transformations for mapplet input, the mapplet is the first object in the mapping pipeline and contains no input groups.

A mapplet must contain at least one Input transformation or Source transformation.

For information about configuring Input and Source transformations, see *Transformations*.

Mapplet output

Mapplet output can be to an Output transformation, a Target transformation, or both.

Use an Output transformation when you want the mapplet to pass data to one or more downstream transformations. When you use the mapplet in a Mapplet transformation, each Output transformation becomes an output group. Each output group can pass data to one or more pipelines in a mapping.

Use a Target transformation when you want the mapplet to write data to a target. When you use a Target transformation with no Output transformation, the mapplet is the last object in the mapping pipeline.

A mapplet must contain at least one Output transformation or Target transformation.

For information about configuring Output and Target transformations, see *Transformations*.

Parameters in mapplets

You can use input parameters in a mapplet. You specify the value of the parameters when you configure the mapping task.

You configure parameters in mapplets the same way that you configure parameters in mappings. For information about configuring parameters, see *Mappings*.

When you include parameters in a mapplet, Data Integration renames the parameters when you use the mapplet in a Mapplet transformation. The parameter names are prefixed with the name of the Mapplet transformation. You can view the corresponding names on the **Parameters** tab of the Mapplet transformation Properties panel.

For example, you use a Filter transformation in a mapplet and select **Completely Parameterized** as the filter condition. You create a string parameter called "Filter_Param." You use the mapplet in a Mapplet transformation named "MyMappletTx." In the Mapplet transformation, the filter parameter is renamed to "MyMappletTx_Filter_Param."

You cannot use in-out parameters in mapplets.

Creating a mapplet

Create a mapplet in the Mapplet Designer.

The Mapplet Designer contains the same design areas and functionality as the Mapping Designer. For information about using the Mapping Designer, see *Mappings*.

1. Click **New** > **Mapplets** > **Mapplet**.

   The Mapplet Designer appears with an Input transformation and an Output transformation on the canvas.
2. Specify the mapplet name and location.

   The default name for the mapplet is *Mapplet* followed by a sequential number.
   If the **Explore** page is currently active and a project or folder is selected, the default location for the asset is the selected project or folder. Otherwise, the default location is the location of the most recently saved asset.

   You can change the name or location after you save the mapping using the **Explore** page.
3. Optionally, enter a description for the mapplet.

   When you use the mapplet in a Mapplet transformation, the description appears in the **Mapplet** tab.
4. Configure the mapplet input.

   Input can be a Source transformation, an Input transformation, or both. If you use an Input transformation, add input fields for the fields you want to pass to the mapplet.
5. Add transformations to the mapplet.
6. Configure the mapplet output.

   Output can be a Target transformation, an Output transformation, or both. If you use an Output transformation, add output fields for the data you want to pass to the downstream transformation.
7. Validate and save the mapplet.

Editing a mapplet

Edit a mapplet in the Mapplet Designer.

When you edit a mapplet, the Mapplet Designer validates the transformation logic. When you save the mapplet, if the mapplet is valid, Data Integration deploys the changes to each mapping and mapplet that uses the mapplet.

Use caution when editing mapplets that are used in mappings and other mapplets. Changes to the mapplet interface result in validation errors in the mappings and mapplets that use the mapplet. To see the assets that use the mapplet, in the Mapplet Designer, open the **Actions** menu and select **Show Dependencies**.

You can make the following changes to a mapplet without affecting the assets that use the mapplet:

- Change transformation names, descriptions, or properties.
- Add or remove transformations in the mapplet, as long as you do not change the mapplet type from active to passive or from passive to active.

The following changes to a mapplet change the interface of the mapplet:

- Add or remove transformations that change the mapplet type from active to passive or from passive to active.

- Change the data type, precision, or scale of a connected input or output field.
- Add or remove input or output fields
- Add or remove Input or Output transformations

If you change the interface of a mapplet, you must synchronize the Mapplet transformations that use the mapplet to get the latest changes. If you do not synchronize the mapplet transformation and you run the task, the task fails.

To edit a mapplet, open the mapplet from the Explore page. You cannot edit a mapplet in a Mapplet transformation.

You cannot edit a mapplet that you imported from PowerCenter.

Synchronizing a mapplet

If the interface of a mapplet changes after it has been added to a Mapplet transformation, you must synchronize the mapplet to get the changes. Synchronize a mapplet on the **Mapplet** tab.

Mappings and mapplets that use the mapplet are invalid until the mapplet is synchronized. If you run a mapping task that includes a changed mapplet, the task fails.

When you synchronize a mapplet, the updates might cause validation errors in other transformations in the mapping or mapplet.

You cannot synchronize a mapplet that you imported into Data Integration from PowerCenter or SAP.

To synchronize a mapplet, perform the following steps:

1. Open the mapping or mapplet that uses the mapplet.
2. Select the mapplet transformation.
3. On the **Mapplet** tab, click **Synchronize**.
4. Correct any resulting errors in the transformation logic.

PowerCenter mapplets

If you are an Informatica PowerCenter user, you can use a PowerCenter mapplet to create a mapplet in Data Integration.

To use a PowerCenter mapplet, you create a mapplet in PowerCenter and export the mapplet to an XML file. Then you upload the XML file in to Data Integration.

A mapplet contains a set of transformations. A PowerCenter mapplet can contain one or more Source transformations but it cannot contain a Target transformation. You can use PowerCenter mapplets in the following Data Integration tasks:

- Synchronization tasks. You can use one mapplet in a synchronization task.
- Mapping tasks. You can use multiple mapplets in a mapping task. You can use mapplets in a Visio template or in a mapping.
- Masking tasks. You can use passive mapplets in a masking task to mask target fields.

Active and passive PowerCenter mapplets

Mapplets are either active or passive.

When you upload a PowerCenter mapplet in Data Integration, you specify whether the mapplet is active or passive.

An active mapplet includes at least one active PowerCenter transformation. An active mapplet can return a number of rows different from the number of source rows passed to the mapplet. For example, an active mapplet might aggregate five source rows and return one row.

A passive mapplet includes only passive PowerCenter transformations. A passive mapplet returns the same number of rows that are passed from the source.

Stored Procedures in mapplets

When a PowerCenter mapplet that you want to use contains a Stored Procedure transformation, the stored procedure must include exception handling.

Exception handling can be as complex as necessary. Or, you can use the following simple example:

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

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

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

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

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

PowerCenter XML files for mapplets

To use a mapplet in Data Integration, you upload a PowerCenter XML file that defines a PowerCenter mapplet.

Consider the following rules when you use a PowerCenter XML file to create a Data Integration mapplet:

- If the mapplet includes a transformation that uses a connection, then the PowerCenter XML file must contain only one workflow, one Session task, one mapping, and one mapplet.
  If the mapplet doesn't include a transformation that uses a connection, then the PowerCenter XML file must include one mapplet. The workflow, Session task, and mapping are optional.
- The session can use any type of connection.
- You do not have to map all source and target fields in the PowerCenter mapping.

- The PowerCenter mapplet can contain the following supported transformations:
  - Aggregator transformation
  - Expression transformation
  - Filter transformation
  - HTTP transformation
  - Lookup transformation
  - Salesforce Lookup transformation (multiple matches returns a single match)
  - Salesforce Picklist transformation
  - Salesforce Merge transformation
  - Sorter transformation
  - Stored Procedure transformation with exception handling
  - Transaction Control transformation
  - Web Services consumer transformation
  - XML Generator transformation with flat file or database sources
  - XML Parser transformation with flat file or database sources
- If you use a mapplet in a synchronization task, the PowerCenter mapplet cannot contain multiple Input transformations.
- If you use a mapplet in a mapping task, the PowerCenter mapplet can contain multiple Input transformations.
- Data Integration flattens PowerCenter mapplets with multiple input groups into mapplets with one input group. Therefore, the ports in each input group in the PowerCenter mapplet must have unique names. If the names are not unique, rename the input ports in PowerCenter before you export the PowerCenter XML file that contains the mapplet.
- The PowerCenter mapplet cannot contain reusable objects such as shortcuts because Data Integration does not use a repository to store reusable objects. Export the mapplet without reusable objects.

PowerCenter mapplets in Data Integration tasks

Use the following rules and guidelines for using PowerCenter mapplets in Data Integration tasks:

- You can add Lookup transformations between the source and a mapplet.
- You can add Expression transformations between the source and a mapplet, and between a mapplet and the target.
- When you add a mapplet to a synchronization task, the synchronization task wizard removes existing field mappings. The synchronization task wizard doesn't remove existing field mappings if you add a passive mapplet between the source and target.
- When you use an active mapplet in a synchronization task that includes a saved query, the synchronization task ignores the configured target option for the task and inserts data to the target.
- Data Integration retains PowerCenter session-level overrides to the mapping.

Configuring a PowerCenter mapplet

To create a mapplet in Data Integration, you upload a PowerCenter XML file that contains a PowerCenter mapplet.

When you upload the PowerCenter XML file, Data Integration creates a mapplet based on the mapplet definition in the XML file.

1. To create a mapplet, click **New** > **Mapplets** > **Mapplet - PC Import** and then click **Create**.

   To edit a mapplet, on the **Explore** page, navigate to the mapplet. In the row that contains the mapplet, click **Actions** and select **Edit**.
2. Configure the following details:

Detail	Description
Mapplet Name	Name of the mapplet. Mapplet names can contain alphanumeric characters, spaces, and the following special characters: _ . + - Maximum length is 100 characters. Mapplet names are not case sensitive.
Location	Project folder in which the mapplet is to reside.
Description	Description of the mapplet. Maximum length is 255 characters.
Mapplet Type	Whether the mapplet is active or passive. The mapplet type depends on the type of PowerCenter transformations in the mapplet: - Active. The mapplet includes one or more active PowerCenter transformations. - Passive. The mapplet includes only passive PowerCenter transformations.

3. To upload the PowerCenter XML file, click **Upload**.
4. In the **Upload Mapplet XML File** dialog box, click **Choose File**.
5. Browse to the appropriate location and select the PowerCenter XML file.
6. Click **OK**.

   The Mapplet XML File Details area shows the connections, input fields, and output fields.
7. Click **Save**.

CHAPTER 8

Saved queries

A saved query is a component that you create to run SQL statements against a database. You can use a saved query as the source object in a synchronization task or as the query in an SQL transformation.

You can use a saved query in the following places:

As the source in a synchronization task

Create a saved query when you want to use a database source that you cannot configure using the single- or multiple-object source options. For example, you might create a saved query to include source filters, or to perform a complicated join of multiple tables. The query must be a SELECT statement.

To use a saved query in a synchronization task, first create the saved query component. Then when you configure the synchronization task, select the saved query as the source object. You can add one saved query to a synchronization task.

As the query in an SQL transformation in a mapping

You can create a saved query to use as the query in an SQL transformation. The query can contain one or more SQL statements. You can use one saved query in an SQL transformation.

To use a saved query in an SQL transformation, first create the saved query component. Then when you configure the SQL transformation in a mapping, you select the saved query to use.

To use saved queries, your organization must have the appropriate license. For more information, contact Informatica Global Customer Support.

Saved query syntax

When you create a saved query, enter an SQL statement that is valid for the database that you want to run the query against.

You can use different SQL statements based on where you use the saved query:

Synchronization task

When you create a saved query to use as the source in a synchronization task, the SQL statement must be a SELECT statement. Data Integration uses the SQL statement to retrieve source column information. You can edit the data type, precision, or scale of each column before you save the saved query.

For example, you might create a saved query based on a TRANSACTIONS table that includes transactions from 2016 with the following SQL statement:

```
SELECT TRANSACTION_ID, TRANSACTION_TOTAL, TRANSACTION_TIMESTAMP from
dbo.TRANSACTIONS WHERE TRANSACTION_TIMESTAMP>'0:0:0:0 01/01/2016'
```

Data Integration ensures that saved query column names are unique. If an SQL statement returns a duplicate column name, Data Integration adds a number to the duplicate column name as follows:

<column_name><number>

SQL transformation

When you create a saved query to use in an SQL transformation, you can use one or more of the following SQL statements in the query:

ALTER

CALL

COMMENT

COMMIT

CREATE

DELETE

DROP

EXPLAIN PLAN

GRANT

INSERT

LOCK TABLE

MERGE

RENAME

REVOKE

ROLLBACK

SELECT

TRUNCATE

UPDATE

Use the following guidelines when you create the query:

- You can use aggregate functions such as COUNT with Salesforce connections only.
- Do not use conversion functions, such as TO_CHAR or TO_DATE.
- Do not use an asterisk (*) to select all columns of a table. List the columns that you want to select.
- Do not use SQL transformation parameter binding or string substitution notation such as ?input_field? or ~input_field~ in the query. Saved queries have no information about SQL transformation input fields.

Tip: Test the SQL statement you want to use on the source database before you create a saved query. Data Integration does not display specific error messages for invalid SQL statements.

Using saved queries in synchronization tasks

Use the following rules and guidelines to use a saved query in a synchronization task:

- You can add one saved query to each synchronization task.
- You cannot delete a saved query that is used in a synchronization task.

- If you edit column information for a saved query, the synchronization task does not write error rows to the error rows file at runtime.
- In the log details for the task, the success and error row counts for sources might not be accurate when you use a saved query.
- When you use an active mapplet with a synchronization task that includes a saved query, the synchronization task inserts data to the target even if you select a different target option.

Using saved queries in SQL transformations

Configure an SQL transformation to use a saved query on the **SQL** tab of the **Properties** panel. You can use one saved query in an SQL transformation.

1. On the **SQL** tab of the **Properties** panel, select the connection that contains the tables that you want to run the query against.
2. Set the SQL type to **SQL Query**.
3. Set the query type to **Saved Query**.
4. Select the saved query that you want the SQL transformation to process.

For more information about SQL transformations, see *Transformations*.

Configuring a saved query

Create a saved query that you can use as a source object in synchronization tasks or as a query in SQL transformations.

1. Click **New** > **Components** > **Saved Query** and then click **Create**.

   To edit a saved query, on the **Explore** page, navigate to the saved query. In the row that contains the saved query, click **Actions** and select **Edit**.
2. Enter the following details:

Detail	Description
Saved Query Name	Name of the saved query. Query names can contain alphanumeric characters, spaces, and the following special characters: _ . + - Maximum length is 100 characters. Query names are not case sensitive.
Location	Location of the saved query. Browse to the folder where you want to store the saved query or use the default location. If the **Explore** page is currently active and a project or folder is selected, the default location for the asset is the selected project or folder. Otherwise, the default location is the location of the most recently saved asset.

Detail	Description
Description	Description of the saved query. Maximum length is 255 characters.
Database Type	Source database type. Select one of the following database types: - Salesforce - Oracle - SQL Server - MySQL - ODBC - MS Access To use the query in an SQL transformation, the database type must be Oracle or SQL Server.
SQL Query	SQL statements that make up the SQL query. To use the query as the source in a synchronization task, enter a SELECT statement. Data Integration uses the SELECT statement to retrieve the source column information. To use the query in an SQL transformation, enter one or more valid SQL statements. Do not use SQL transformation parameter binding or string substitution notation in the query because saved queries have no information about SQL transformation input fields.

3. If you enter a SELECT statement, click **Get Columns** and select a connection.

   The Saved Query Column Details table displays the columns selected in the SQL statement.
4. Optionally, in the Saved Query Column Details table, edit the data type, precision, or scale.

   If you edit these values, Data Integration does not write error rows into the error rows file.
5. Click **Save**.

CHAPTER 9

Shared sequences

Shared sequences are reusable sequences that you can use in multiple Sequence Generator transformations. When you use a shared sequence, the Sequence Generator transformation uses the properties of the shared sequence to generate values. Use a shared sequence when you want to assign numeric values within the same sequence to data in multiple mapping tasks.

Multiple mappings and mapplets can use the same shared sequence. When you run the mapping task, Data Integration reserves a set of values in the sequence so that each mapping task generates unique values.

For example, you want to assign a unique ID to entries in your customer tables. You have two mappings that load customer data to the same target table, and the mappings are scheduled to run at the same time. Use a shared sequence to ensure that you do not get duplicate IDs in the target.

You create shared sequences on the **New Assets** page. You can create non-shared sequences in a Sequence Generator transformation. For more information about creating non-shared sequences, see *Transformations*.

Shared sequence properties

When you create a shared sequence, you define general properties such as the sequence name, and properties that define the sequence such as initial value, increment value, and end value.

The following table lists the properties you define for a shared sequence:

Property	Description
Name	Required. Name of the shared sequence. Can contain alphanumeric characters and underscores (_). Maximum length is 200 characters.
Location	Project or folder in which the sequence resides. If the **Explore** page is currently active and a project or folder is selected, the default location for the asset is the selected project or folder. Otherwise, the default location is the location of the most recently saved asset.
Description	Optional. Description for the shared sequence.
Increment By	The difference between two consecutive values in a generated sequence. For example, if Increment By is 2 and the existing value is 4, then the next value generated in the sequence will be 6. Default is 1. Maximum is 2,147,483,647.

Property	Description
End Value	Maximum value that the mapping task generates. If the sequence reaches this value during the task run with rows left to process and the sequence is not configured to cycle, the run fails. If the sequence is configured to cycle, the sequence uses this value and then starts over at the Cycle Start Value. Maximum is 9,223,372,036,854,775,807.
Initial Value	The value you want the mapping task to use as the initial value in the sequence the first time that the sequence is used. If you want to cycle through a series of values, this value must be greater than or equal to the Cycle Start Value and less than the End Value. If you reset the shared sequence, the sequence is reset to this value. Default is 0.
Cycle	If enabled, the mapping task cycles through the sequence range. If disabled, the task stops the sequence at the configured End Value. The session fails if the task reaches the End Value and still has rows to process. Default is disabled.
Cycle Start Value	Start value of the generated sequence that you want the mapping task to use when the sequence is configured to cycle. When the sequence reaches the end value, it cycles back to this value. Default is 0.
Number of Reserved Values	The number of sequence values the mapping task caches at one time. Must be greater than or equal to 1000.
Current Value	Displays the current start value in the sequence.

Number of reserved values

The number of reserved values determines the number of values in the sequence that the mapping task caches at one time during each task run.

When multiple Sequence Generator transformations use the same shared sequence, multiple instances of the shared sequence can exist at the same time. To avoid generating duplicate values, reserve a range of sequence values for each mapping by configuring the number of reserved values.

For example, you configure a shared sequence as follows: Number of Reserved Values = 1000, Initial Value = 1, Increment By = 1. Two mappings use the shared sequence. When the first mapping task runs, Data Integration reserves 1000 values (1-1000) for the mapping task and updates the current value in the repository to 1001. When the second mapping task runs, Data Integration reserves the next 1000 values (1001-2000) and updates the current value to 2001. When either mapping task uses all of its reserved values, Data Integration reserves a new set of values and updates the current value.

The number of values that Data Integration reserves in a sequence is determined by multiplying the Number of Reserved Values by the Increment By value.

For example, if you have a shared sequence that begins at 2, increments by 2 and the number of reserved values is 1000, Data Integration reserves 1000 values in the sequence, or numerical values 2-2002, and updates the current value to 2003.

By default, the number of reserved values is 1000. To increase performance, you can increase the number of reserved values. This reduces the number of calls that Data Integration must make to the repository. The number of reserved values cannot be less than 1000.

Values that are reserved for a task run but not used in the task are lost when the task completes. For example, you generate IDs in a table with 1250 rows. You configure the shared sequence to generate values in increments of 1 and set the number of reserved values to 2000. When the task runs, Data Integration reserves 2000 values for the task and updates the current value to 2001. When the next task runs, the sequence begins at 2001, and values 1251-2000 are lost.

Note: If you use partitions, Data Integration reserves a set of values for each partition.

Creating a shared sequence

Create a shared sequence on the **New Asset** page.

1. Click **New** > **Components** > **Shared Sequence** and then click **Create**.
2. Define the sequence properties.
3. Save the sequence.

Using a shared sequence

You can use a shared sequence in a Sequence Generator transformation in a mapping or mapplet.

1. Open the mapping or mapplet.
2. Add a Sequence Generator transformation to the canvas and connect it to a downstream transformation.
3. On the **Sequence** tab, select **Use Shared Sequence**.
4. Select the sequence to use.
5. Configure the Sequence Generator advanced properties and output fields.
   For more information about configuring a Sequence Generator transformation, see *Transformations*.

Note: If you do not select **Use Shared Sequence**, the Sequence Generator transformation generates a non-shared sequence.

Resetting a shared sequence

Reset a shared sequence to update the current value of the sequence to the defined initial value.

1. Open the shared sequence.
2. Click **Edit**.
3. Click **Reset Current Value** in the **Sequence Properties** area.
4. Save the shared sequence.

CHAPTER 10

User-defined functions

User-defined functions are reusable functions that you can use in expressions. Create user-defined functions to build complex expressions using the Informatica Intelligent Cloud Services transformation language. User-defined functions use the same syntax and can use the same transformation language components as transformation and field expressions.

You can include a user-defined function in a transformation expression in a mapping or mapplet, in a field expression in a mapping task, or in another user-defined function. To use a user-defined function in an expression, you select the user-defined function in the expression editor and enter the required arguments.

You cannot use a user-defined function in an expression in a synchronization task. You also cannot use a user-defined function in any transformation in an elastic mapping.

To create and use user-defined functions, your organization must have the appropriate license.

Example

You want to remove leading and trailing spaces from a text string such as a name or address. You create a user-defined function called RemoveSpaces that takes a text string as an argument and performs the LTRIM and RTRIM functions. When you configure the user-defined function, you enter the following expression:

```
LTrim( RTrim(TextString) )
```

After you create the function, you use it in an Expression transformation to remove leading and trailing spaces from the incoming field, LAST_NAME. The expression field contains the following expression:

```
:UDF.RemoveSpaces(LAST_NAME)
```

In the expression, the user-defined function name is prefaced with :UDF. The incoming field LAST_NAME is passed to the function as the argument.

Creating user-defined functions

Create a user-defined function on the **New Asset** page. The user-defined functions that you create are available to all users in your organization based on their privileges.

1. Click **New** > **Components** > **User-Defined Function** and then click **Create**.
2. Configure general properties such as the function name, location, and return type.
3. Optionally, create arguments for the function.

   When you create arguments, configure the name, data type, and description for each argument.
4. Create the function expression.
5. Validate and save the function.

User-defined function general properties

When you create a user-defined function, you must define general properties such as the function name and return type. Define general properties on the **General** tab.

The following table describes the properties:

Property	Description
Name	Name of the function. Must be unique within an organization. The name must begin with a letter and can contain letters, numbers, and the following special characters: `_ @ $ #` The name cannot exceed 100 characters and cannot contain spaces.
Location	Location of the user-defined function. Browse to the folder where you want to store the user-defined function or use the default location. The default location is the location of the most recently saved asset.
Description	Description of the user-defined function. The description appears in the expression editor when you select the function in a transformation expression, a field expression, or another user-defined function.
Return Type	Data type of the values that the function returns. Can be binary, date, numeric, or string.

User-defined function arguments

When you create a user-defined function, you can include arguments to define the values that you pass to the function. Create arguments on the **Arguments** tab. You can create up to 10 arguments.

To create an argument, click **Add**, and enter the following information:

Property	Description
Name	Name of the argument. Must be unique within the function. The name must begin with a letter and can contain letters, numbers, and underscore characters (_). It cannot exceed 100 characters and cannot contain spaces.
Type	Data type of the argument. Can be binary, date, numeric, or string.
Description	Description of the argument. The description appears in the Arguments and Functions area on the **Expression** tab when you select the argument. It also appears in the expression editor when you use the function in a transformation or field expression.

Arguments are passed to the function in the order in which they appear on the **Arguments** tab. To rearrange the order, select an argument and click **Move up** or **Move down**.

To delete an argument, click **Delete** in the row that contains the argument.

User-defined function expression

Create the expression that defines the function on the **Expression** tab. The function expression can contain constants, operators, built-in functions, and other user-defined functions. You can create a complex expression by nesting functions within functions.

To create an expression, enter the expression in the expression editor. The expression that you create must follow the same guidelines and use the same syntax as any transformation or field expression. For more information about creating expressions, see *Tasks*.

You can add arguments, built-in functions, and other user-defined functions to the expression by selecting them in the Arguments and Functions area of the expression editor and clicking **Add**. You can also type in the expression manually.

An expression cannot contain cyclic references. For example, the expression for user-defined function Function_A cannot include Function_A. Also, if user-defined function Function_A calls user-defined function Function_B, then the expression for Function_B cannot call Function_A.

To validate the expression, click **Validate**. Data Integration validates the expression. It does not validate the user-defined functions and expressions that use the function.

Editing and deleting user-defined functions

Certain restrictions apply when you edit or delete a user-defined function.

Editing a user-defined function can affect the expressions and functions that use it. Consider the following guidelines when you edit a user-defined function:

- If you edit a user-defined function and the updated function is valid, Data Integration propagates the changes to all expressions and user-defined functions that use the function.

  Sometimes this can make the expressions and functions that use the user-defined function invalid. For example, you edit a function, change the number of arguments from one to two, and validate the function. Even though the updated function is valid, expressions and functions that use the user-defined function become invalid if they pass one argument to the function.
- If you edit and save a user-defined function but the updated function is not valid, Data Integration does not propagate the changes to the expressions and user-defined functions that use it.
- You cannot rename a user-defined function after you save it.

To delete a user-defined function, you must first remove it from all expressions and functions that use it. You cannot delete a user-defined function that is used in an expression or in another user-defined function.

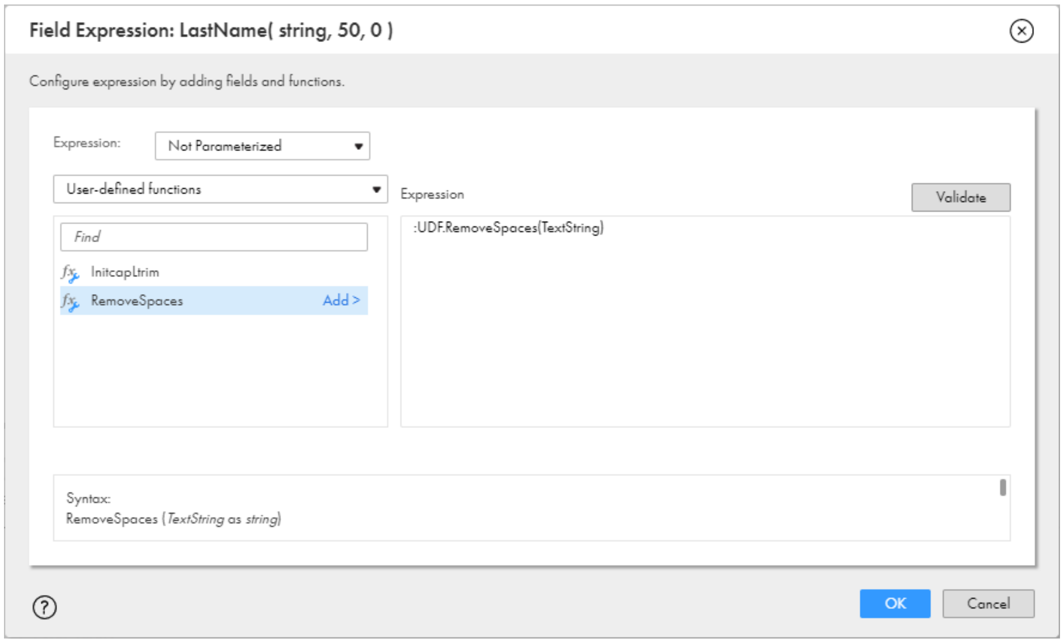
Tip: View the dependencies for the user-defined function to find the assets that use the function.

Creating expressions with user-defined functions

You can add a user-defined function to a transformation or field expression.

When you create an expression, valid user-defined functions appear in the expression editor. If you type in an expression manually, precede each user-defined function with :UDF.

The following image shows a user-defined function selected in the expression editor of an Expression transformation:



:UDF.RemoveSpaces(TextString)

Syntax:
RemoveSpaces (*TextString as string*)

When you select a user-defined function, the expression editor shows the function syntax in the following format:

```
<function name> (<argument 1> as <data type>, <argument N> as <data type>)
```

For example:

```
RemoveSpaces(TextString as string)
```

When you add the function to the expression, the function includes the prefix :UDF, as shown in the following example:

```
:UDF.RemoveSpaces(TextString)
```

After you add the function to the expression, replace the arguments with field names or parameters. For more information about creating expressions, see *Tasks*.

When you validate the expression, Data Integration does not validate the user-defined function. It only validates the expression.

INDEX