IBM InfoSphere DataStage and Quality Stage
Version 11 Release 3

Guide to Accessing Unstructured Data

SC19-4330-00

IBM InfoSphere DataStage and Quality Stage
Version 11 Release 3

Guide to Accessing Unstructured Data

SC19-4330-00

Note

Before using this information and the product that it supports, read the information in “Notices and trademarks” on page 63.

Contents

Chapter 1. Unstructured Data Stage

Unstructured data is information that does not have a predefined data model or does not fit well into relational tables. Unstructured data can be text from books, journals, metadata, audio, video files, the body of word processor documents, web pages, and presentation charts. In this release, the Unstructured Data stage supports only Microsoft Excel files as data sources.

Use the Unstructured Data stage to perform the following operations:

- Extract information from unstructured data sources and integrate the information with your jobs. Data from Microsoft Excel sheets that has different column definitions can be extracted from a single Unstructured Data stage.
- Write data to Microsoft Excel sheets.

Designing jobs with Unstructured Data stages

You can use Unstructured Data stages in your jobs to read data from Unstructured Data sources or write data to Unstructured Data sources in the contexts of those jobs.

Defining a job that includes a Unstructured Data stage

Before you can read or write data from or to a Microsoft Excel files, you must create a job that includes the Unstructured Data stage, add any required additional stages, and create the necessary links.

Procedure

1. From the Designer client, click **File** > **New**.
2. In the New window, click the **Parallel Job** icon, and then click **OK**.
3. From the Palette, click **File**.
4. Drag the **Unstructured Data stage** icon to the canvas.
5. Create stages for the job.
6. On the left side of the Designer client in the Palette menu, select the **General** category, and then create the necessary links for the job.
7. (Optional) Double click the **Unstructured Data stage** icon to enter or modify the following attributes:
   - **Stage :** Modify the default name of the **Stage**. You can enter up to 255 characters. Alternatively, you can modify the name of the stage in the job design canvas.
   - **Description:** Enter an description of the stage.
8. Click **Save**.

Extracting the data from Microsoft Excel

You can use the Unstructured Data stage to extract several types of data from a Microsoft Excel file.

Prerequisites

Before you start the installation and configuration of Unstructured Data stage, make sure that you meet the system requirements and that you have installed all the prerequisite software.

Before you begin

Procedure

1. Install Information Server of language that matches the language of Microsoft Excel file that you want to extract.
2. Ensure that the Microsoft Excel Viewer program that shows the content of Microsoft Excel spreadsheets (for example, Microsoft Excel, Microsoft Excel Viewer or IBM Lotus Symphony) is installed on your client machine.
3. Ensure that the file extension `.xls` and `.xlsx` are properly associated to your Microsoft Excel Viewer program.

Supported data sources

The Unstructured Data stage supports only Microsoft Excel files as the source file.

The Unstructured Data stage supports the following file formats:

- Microsoft Excel 97-2003 OLE2 (`.xls`), including support of password-encrypted file.
- Microsoft Excel 2007-2010 OOXML (`.xlsx`), including support of password-encrypted file.

The Unstructured Data stage does not support Microsoft Excel files that are created by Microsoft Excel for Mac.

Data ranges

When you use the Unstructured Data stage, you can extract data from a specified data range in a Microsoft Excel spreadsheet.

Data range represents a cell, a row, a column, or a selection of cells that contain one or more continuous blocks of cells. A data range is specified by the range expression. In the Unstructured Data stage, you can use a range expression to specify the data range to extract.

For example, Employee_Salary!A1:G8 describes a data range in which the first cell is A1 and the last cell is G8 in the Employee_Salary spreadsheet.

Table 1. Example of Microsoft Excel file: Employee_Salary spreadsheet

	A	B	C	D	E	F	G
1	**EMPNO**	**FIRSTNAME**	**LASTNAME**	**DEPT**	**JOB**	**SALARY**	**BONUS**
2	20	MICHAEL	THOMPSON	B01	MANAGER	94250	800
3	30	SALLY	KWAN	C01	MANAGER	98250	800
4	60	IRVING	STERN	D11	MANAGER	72250	500
5	70	EVA	PULASKI	D21	MANAGER	96170	700
6	50	JOHN	GEYER	E01	MANAGER	80175	800
7	90	ELEEN	HENDERSON	E11	MANAGER	89750	600
8	100	THEODORE	SPENSER	E21	MANAGER	86150	500

The Unstructured Data stage maps the Microsoft Excel row and column in the specified data range to InfoSphere® DataStage® row and column, and extracts the records.

The following table describes the records extracted by the Unstructured Data stage when the range expression is Employee_Salary!A2:G8.

Table 2. Example of DataStage row and column

20	MICHAEL	THOMPSON	B01	MANAGER	94250	800
30	SALLY	KWAN	C01	MANAGER	98250	800
60	IRVING	STERN	D11	MANAGER	72250	500
70	EVA	PULASKI	D21	MANAGER	96170	700
50	JOHN	GEYER	E01	MANAGER	80175	800
90	ELEEN	HENDERSON	E11	MANAGER	89750	600
100	THEODORE	SPENSER	E21	MANAGER	86150	500

The **Range option** property of Unstructured Data stage allows you to specify the data range either by selecting the **Specify the start row** option or the **Specify the entire data range** option. If you select the **Specify the start row** option, then identify the start row. Unstructured Data stage then identifies the end row of the data range. If you select the **Specify the entire data range** option, then you must specify the start and end rows of the data range to be extracted.

If you want to use the value of cells in the first row as IBM® InfoSphere DataStage column name, then you can use the **Column header** property. If the **Column header property** is set to **First row of data ranges**, and if you specify the range expression as Employee_Salary!A1:G8, the first row is treated as header, and the value of the cells in the first row is used as default DataStage column name in the job. You can generate range expression at design time by using Unstructured Data stage.

Types of data that can be extracted from Microsoft Excel

You can use the Unstructured Data stage to extract several types of data from a Microsoft Excel file.

File properties

The following table lists the information that can be extracted as file properties:

Table 3. Data that can be extracted as file properties

Data	Description
File name	Name of the file. For example: `Workbook1.xls`
File path	Path of the file. For example: `C:\excel\Workbook1.xls`
File size	Size of the file in bytes.
Last modified date	The date and time that the file was last modified.

Document properties

The following table lists the information that can be extracted as document properties:

Table 4. Data that can be extracted as document properties

Data	Description
Authors	Authors of the document.

Table 4. Data that can be extracted as document properties (continued)

Data	Description
Document comments	Comments of the document.
Content creation date	The date and time that the document was created.
Key words	Key words of the document.
Revision number	Revision number of the document.
Subject	Subject of the document.
Title	Title of the document.
Company	Company property value of the document.
Category	Category of the document.
Manager	Manager of the document.
Custom properties	Custom properties of the document. You must specify the name of the custom property to extract.

Sheet information

The following table lists the information that can be extracted as sheet information:

Table 5. Data that can be extracted as sheet information

Data	Description
Sheet name	Name of the Microsoft Excel sheet.
Header (left, center, right)	Header of the specified position.
Footer (left, center, right)	Footer of the specified position.

Row information

The following table lists the information that can be extracted as row information:

Table 6. Data that can be extracted as row information

Data	Description
Row number	Microsoft Excel row number within the sheet. The first row number is 1.
Is hidden	Whether the row is hidden or not. Writes true if the row or the sheet to which this row belongs is hidden.

Cell information

You can extract the cell information based on the Microsoft Excel column or the cell position. You can specify the source Microsoft Excel column based on the relative position within the data range when extracting the cell information based on the Microsoft Excel column.

The following table lists information that can be extracted as cell information:

Table 7. Data that can be extracted as cell information

Data	Description
Value	Value of a cell. If the cell has a formula, the stage extracts the value from the cache.
Comment	Comment of a cell.
Author of Comment	Author of the comment of a cell.
Formula	Formula of a cell in text.
Hyperlink Type	Type of hyperlink of a cell.
Hyperlink Address	The address this hyperlink points to. The format depends on type of this hyperlink.
Hyperlink label	Text label for this hyperlink.

Designing jobs that extract data from Microsoft Excel file

You can use Unstructured Data stage to design jobs that read unstructured data from Microsoft Excel files.

About this task

The Unstructured Data stage reads data from the Microsoft Excel files and passes the rows to a Transformer stage. The Transformer stage transforms the data and then loads the data into the ODBC connector. When you configure the Unstructured Data stage to read data from the Microsoft Excel files, you create only one output link.

Procedure

1. Define a job that includes a Unstructured Data stage.
2. To set up the Unstructured Data stage as a source stage to read unstructured data from Microsoft Excel files, complete the following steps:
   a. Configure the Unstructured Data stage as a source.
   b. Modify the column definition on the link.
3. Compile and run the job.

Configuring the Unstructured Data stage as a source:

When you create an Unstructured Data stage job, you must configure the Unstructured Data stage so that it extracts the data and generates the output in the data type that the user requires.

Procedure

1. On the parallel canvas, double-click the **Unstructured Data** stage.
2. From the **Document Type** list, select **Excel**.
3. Click **Configure** to configure additional properties, and define mapping between Microsoft Excel items and DataStage columns.
4. Specify the file name details in the Data source pane:
   a. Specify the name of the file from which you want to read the data, in the**File name** field. Job compilation fails if this field is empty. If the file is password protected, specify password in the **Password** field.
   b. Optional: If you specify wildcard characters in the file name, select **Use template file for design time** and specify a template file name. Template

file is used for subsequent configuration steps, and not used at runtime. Specify a value for **Template password** if the specified template file is password protected.

c. Optional: Click **View** to launch the external Microsoft Excel viewer program. You can confirm the content of Microsoft Excel file you are working with.

d. Click **Load**.

5. Specify the data range details to read from the Microsoft Excel file in the Read options pane.

   a. Optional: Specify a value for **Range option**. If you select **Specify the start row**, you only need to specify the first row. Unstructured Data stage finds the last row at runtime. If you select **Specify the entire data range**, you need to specify both start row and end row.

   b. Optional: Specify **Range expression**. **Range expression** is a required property at runtime, but it can be empty when clicking **Load** button. Unstructured Data stage searches the entire document and lists the candidates of data range in the **Template data range** list box in the **Import** pane. **Range expression** property is set with the appropriate value when you click **Import** in the **Import** pane.

   c. Optional: If you want to skip any sheet names from range expression, then specify the name in the **Specify the Sheet names to skip** field. Use this field when the sheet names are omitted from the range expression.

   d. Optional: Specify First row of data ranges. At design time, if you select **None**, Microsoft Excel column names are expressed in the format: "Column#*column number*(Column*Excel column label*)" in the Map pane. If you select **First row is header** , then the first row value is displayed in the Map pane. At runtime if you select **None**, the first row is extracted. If you select **First row is header**, the first row is skipped.

6. Specify data range details to import in the Import pane.

   a. Select one data range from **Template data range**.

   b. Optional: If you want to extract additional Microsoft Excel items such as document properties, select **Property** tab and select items to be extracted.

   c. Click **Import**.

7. Define the mapping between InfoSphere DataStage columns and Microsoft Excel items in the Map pane.

   a. Define the mapping between InfoSphere DataStage columns and imported Microsoft Excel items. You can add InfoSphere DataStage column mappings or change the column order by clicking **Up**, **Down**, **Insert**, or**Delete** buttons. In **InfoSphere DataStage Column**, specify the InfoSphere DataStage column name for each Microsoft Excel item. In **Microsoft Excel Item**, you can select the item you want to map to the InfoSphere DataStage column. All items that can be selected in the Import pane are listed in each cell. In **Import Option**, you can select the Microsoft Excel item if there is any import options available. For example, If you select Microsoft Excel column in Excel Item, Value, Comment, Author of Comment, Formula, Hyperlink Type, and Hyperlink Address options are available.

   b. Click **OK**.

8. Specify required details in the **Properties** tab and the **Advanced** tab.

9. Click **OK** to save the settings that you specified. to save the settings that you specified.

Modifying the column definition on the link:

You can modify the column definition such as SQL Type, Length, Scale, and Nullable on the link. If you want to change the column name that was imported by **Configuration** window, launch the **Configuration** window again to specify the name.

Procedure

1. On the parallel canvas, double-click the **Unstructured Data** stage.
2. Select the **Output** tab, then select the output link from **Output name (downstream stage)**.
3. Edit the SQL type, Length, and Scale of each column.
4. Click **OK** to save the changes.

Using job parameters

Unstructured Data stage does not have the ability to create new job parameters in Configuration window. However, you can use the job parameters in the Configuration window. You must create job parameters in the Job Properties window before or after you work on the Configuration window, by selecting **Edit > Job Properties** from IBM InfoSphere DataStage and QualityStage® Designer client. For more information about creating job parameters, see Lesson 2.4: Adding parameters in the IBM InfoSphere DataStage Parallel Job Tutorial.

A job parameter is specified in the Configuration Window with a **#** character. For example, job parameter *FileName*, is specified as **#FileName#** in the Configuration window. For String type field such as **File name** property, you can directly type the name of job parameter within **#**.

If you want to use job parameter for the List type property such as **Range option**, you must create a List type parameter that contains a list of string variables. The String variables must match with the label text of the corresponding property in the Configuration window. For example, if you want to use job parameter for **Range option** property, you must create a List type job parameter that contains the string variable **Specify the start row** and **Specify the entire data range**. After creating a job parameter, select **<Parameterize...>** from the Configuration window, and specify the job parameter name within the **#** character in the **Input Parameter** dialog box. Click **Load** to edit or select variables in the Resolve job parameters panel.

Options to read data from Microsoft Excel files

Use the following options to modify how the Unstructured Data stage reads data.

Error handling:

You can specify whether to log an error message and stop the job when an error occurs while extracting data from the file.

You can set the **Error action** property to **Fail** or **Skip**. The default value is **Fail**.

- If you select **Fail**, the Unstructured Data stage logs an unrecoverable error and stops the job when an error occurs while extracting data.
- If you select **Skip**, the Unstructured Data stage logs a warning message and continues to process the remaining input fields and records when an error occurs while extracting data.

Null row handling:

You can configure the Unstructured Data stage to skip rows with null values in its cells that are being extracted.

You can set the **Skip null rows** property to **Yes** or **No**. The default value is **No**.

- To skip rows with null values in its cell, select **Yes**.
- To continue with rows with null values in its cells, select **No**.

Extracting the value of a particular cell or custom properties:

You can specify the value of a particular cell or the custom properties to be extracted.

Procedure

1. On the job design canvas, double-click the **Unstructured Data stage** icon.
2. Click **Configure**.
3. From the **Import** pane, select **Advanced** tab.
4. To import the value of a particular cell, select **Particular Cell** in the **Type** column and specify the cell position in the **Value** column. For example, if you want to import the value of cell A1, enter A1 in the **Value** column.
5. To import custom property, select **Custom Property** in the **Type** column and specify the property value in the **Value** field. For example, if you want to import the custom property named as Prop1, specify Prop1 in the **Value** field.
6. Click **Import**.
7. In the Map pane, define mapping between InfoSphere DataStage columns and Microsoft Excel items.
8. Click **OK**.

Propagating hidden columns:

You can specify the action that needs to be taken for the hidden columns during runtime column propagation.

You can set the **Hidden columns** property in the **Runtime Column Propagation** category to either extract or skip the hidden columns during runtime column propagation.

- If you select **Extract**, the hidden columns are extracted.
- If you select **Skip**, the hidden columns are skipped.

The default value is **Extract**.

Runtime column propagation

In InfoSphere DataStage, you can configure a job to propagate extra columns that are not defined in the metadata through the rest of the job. This process is known as runtime column propagation (RCP).

When runtime column propagation is enabled, the Unstructured Data stage propagates Microsoft Excel columns based on the first data range. If wildcard characters are used in the file name, the first file that matches the expression is used. The setting of the hidden columns property determines whether a hidden column is propagated. For each propagated Microsoft Excel column, only cell

values are extracted. To extract information such as the file name, sheet name, or the row number, you can define the additional columns in the configuration window.

Column naming rules

InfoSphere DataStage columns are named based on the Microsoft Excel column letter of the first data range. The column name is prefixed by **"Column_"** followed by the Microsoft Excel column letter. For example, Column_A, Column_B, Column_C, and so on.

If the job already has a column with the name, the job aborts.

Data types

All columns that are added by the Unstructured Data stage are in **Unicode Varchar** type with undefined length.

Examples of extracting data from Microsoft Excel files

You can build sample jobs that extract data from Microsoft Excel files.

To get the files for the examples, extract the `IS_install\Clients\Samples\Connectors\UnstructuredData_Samples.zip` file.

Example 1: Extracting data from a range in an Microsoft Excel file:

Create a job that uses the Unstructured Data stage to retrieve data from a range in an Microsoft Excel spread sheet.

About this task

This example uses the sample Microsoft Excel file, `Employee1.xls`, which contains details of employees working in an organization. This sample file has three spread sheets, `Sheet1`, `Sheet2` and `Sheet3`.`Sheet1` contains information about the employees in every department in the organization. `Sheet2` and `Sheet3` are blank. In this example, you extract business information about only the employees who work for department B01.

Step 1: Creating the job:

Create an example job that includes one Unstructured Data stage and one Sequential File stage.

Procedure

1. Start the IBM InfoSphere DataStage and QualityStage Designer client.
2. In the **Repository** pane, right-click the **Jobs** folder, and then click **New** > **Parallel job**.
3. From the **File** section of the palette, drag the Unstructured Data stage to the canvas.
4. Drag a Sequential File stage to the canvas, then position the stage to the right of the Unstructured Data stage.
5. Create a link from the Unstructured Data stage to the sequential file stage.
6. Rename the stage and the link.
7. Select **File** > **Save**, and name the job.

Step 2: Configuring the Unstructured Data stage:

Configure the Unstructured Data stage to extract data from a range in an Microsoft Excel file.

Procedure

1. Double-click Unstructured Data stage.
2. Click **Configure**.

   Note: Do not configure any stage properties in this step because you can configure all the required configurations in the Configuration window.
3. In the Configuration window, specify the full file path of the Microsoft Excel input file `Employee1.xls`.
4. From the **Range option** list, select **Specify the entire data range**, to extract the data in a specific range.
5. In the **Range expression** field, specify `Sheet1!A16:K28`.
6. From the **Column header** field, select **First row of data ranges**. When **First row of data ranges** is selected, first row is regarded as the header and the Unstructured Data stage starts extracting from the second row.
7. Click **Load**, then make sure that check boxes next to the Microsoft Excel columns to be extracted with the job are selected. The Unstructured Data stage accesses the specified file and lists the Microsoft Excel columns in the specified data range in the Import pane. By default, all Microsoft Excel columns are selected.
8. Clear the check box next to the Microsoft Excel columns E (PHONE NO) and I (BIRTH DATE).
9. Click **Import**. When **Import** is clicked, the Map pane at the lower right of the Configuration window is updated.
10. Click **OK**.
11. Confirm that the values that you specified in the Configuration window are saved on the property tab of the stage editor.
12. In the **Output** > **Column** page, change the type of the **EMP_NO** column to `Integer`, and then click **OK**.

Step 3: Configuring the Sequential File stage:

Configure the Sequential File stage.

About this task

In this example, Sequential File stage is used as output stage. You can use any other output stage for creating the output.

Procedure

1. Double-click the Sequential File stage.
2. On the **Properties** page, specify the path where you want the output file to be created, followed by the file name `OutputOfExample1.txt`.
3. Click **OK**.

Step 4: Viewing the output of the job:

After you run the job, open the file, and verify the output.

About this task

For example, if a Microsoft Excel input file contains the employee information for different departments in an organization, the job can extract data from the specified department.

Procedure

1. Save the job.
2. Compile and run the job.

   The following table displays the information in a Microsoft Excel input file that contains the employee information for different departments.

Table 8. Sample Microsoft Excel file with employee details

EMP NO	FIRST NAME	MID INIT	LAST NAME	PHONE NO	HIRE DATE	JOB	SEX	BIRTH DATE
Employees in DEPT_A00								
10	CHRISTINE	I	HAAS	3978	1/1/1995	PRES	F	8/24/1963
20	MICHAEL	L	THOMSON	3476	10/10/2003	MANAGER	M	2/2/1976
30	SALLY	A	KWAN	4738	4/5/2005	MANAGER	F	5/11/1971
50	JOHN	B	GEYER	6789	8/17/1979	MANAGER	M	9/15/1955
Employees in DEPT_B01								
60	IRVING	F	STERN	6423	9/14/2003	MANAGER	M	7/7/1975
70	EVA	D	PULASKI	7831	9/30/2003	MANAGER	F	5/26/2003
90	EILEEN	W	HENDERSON	5498	8/15/2000	MANAGER	F	5/15/1971
100	THEODORE	Q	SPENSER	742	6/19/2000	MANAGER	M	12/18/1980
110	VINCENZO	G	LUCCHESSI	3490	5/16/1988	SALESREP	M	11/5/1958
120	SEAN		O'CONNELL	2167	12/51993	CLERK	M	10/18/1972
130	DELORES	M	QUINTANA	4578	7/28/2001	ANALYST	F	9/15/1955
140	HEATHER	A	NICHOLLS	1793	12/15/2006	ANALYST	F	1/19/1976
150	BRUCE		ADAMSON	4510	2/12/2002	DESIGNER	M	5/17/1972
160	ELIZABETH	R	PIANKA	3782	10/11/2006	DESIGNER	F	4/12/1980
1770	MASATOSHI	J	YOSHIMURA	2890	9/15/1999	DESIGNER	M	1/5/1981
180	MARILYN	S	SCOUTTEN	1682	7/7/2003	DESIGNER	F	2/21/1978

3. After the job runs, open the `OutputOfExample1.txt` file to view the result.

   The `OutputOfExample1.txt` file displays data for DEPT_B01 except the PHONE NO and the BIRTH DATE columns:

```
"60","IRVING","F","STERN","2003-09-14","MANAGER ","M","72250","500"
"70","EVA","D","PULASKI","2005-09-30","MANAGER ","F","96170","700"
"90","EILEEN","W","HENDERSON","2000-08-15","MANAGER ","F","89750","600"
"100","THEODORE","Q","SPENSER","2000-06-19","MANAGER ","M","86150","500"
"110","VINCENZO","G","LUCCHESSI","1988-05-16","SALESREP","M","66500","900"
"120","SEAN"," ","O'CONNELL","1993-12-05","CLERK   ","M","49250","600"
"130","DELORES","M","QUINTANA","2001-07-28","ANALYST ","F","73800","500"
"140","HEATHER","A","NICHOLLS","2006-12-15","ANALYST ","F","68420","600"
"150","BRUCE"," ","ADAMSON","2002-02-12","DESIGNER","M","55280","500"
"160","ELIZABETH","R","PIANKA","2006-10-11","DESIGNER","F","62250","400"
"170","MASATOSHI","J","YOSHIMURA","1999-09-15","DESIGNER","M","44680","500"
"180","MARILYN","S","SCOUTTEN","2003-07-07","DESIGNER","F","51340","500"
```

Example 2: Extracting data from multiple Microsoft Excel sheets:

Create a job that uses the Unstructured Data stage to extract data from multiple Microsoft Excel sheets.

About this task

This example uses the sample Microsoft Excel file, `Employee2.xls`. This sample file has the following sheets: `DEPT A00`, `DEPT B01`, `DEPT C01`, and `DEPT D01`. Each sheet contains information about the employees in the department.

The data structure of each sheet is similar. Each sheet has the EMP NO, FIRST NAME, MID INIT, LAST NAME, PHONE NO, HIRE DATE, JOB, and ADDRESS columns, and the third row is the header. But each sheet has a different number of rows.

Step 1: Creating the job:

Create an example job that includes one Unstructured Data stage and one Sequential File stage.

Procedure

1. Start the IBM InfoSphere DataStage and QualityStage Designer client.
2. In the Repository pane, right-click the **Jobs** folder, and select **New** > **Parallel job**.
3. From the **File** section of the palette drag an Unstructured Data stage to the canvas.
4. From the **File** section of the palette drag a Sequential File stage to the canvas. Position the stage to the right of the Unstructured Data stage.
5. Create a link from the Unstructured Data stage to the sequential file stage.
6. Rename the stages and links.
7. Select **File** > **Save**, and name the job.

Step 2: Configuring the Unstructured Data stage:

Configure the Unstructured Data stage to extract the data from the multiple Microsoft Excel sheets.

Procedure

1. Double-click the Unstructured Data stage to open the stage properties.
2. Click **Configure**.

   Note: Do not configure any stage properties in this step because you can configure all the required configurations in the Configuration window.
3. In the Configuration window, specify the full file path of the Microsoft Excel input file `Employee2.xls`.
4. From the **Range option**, select **Specify the start row**.
5. In the **Range expression** field, specify **A3:H3** . When the stage runs with Specify the first row option and no specific sheet name is specified in the **Range expression**, the job finds the last row dynamically and extracts rows to the last row at runtime.
6. In **Column header**, select **First row of data ranges**.
7. Click **Load**. The Excel columns that exist in the specified data range are listed in the Import pane.
8. On the **Property** tab, select the checkbox next to the property, to extract the property value. In this example, select the **Sheetname** as the property.
9. Click **Import**. The column mappings are generated by the stage.

10. To make the SheetNname column the first column in the list:
    a. Select the SheetName column.
    b. Click **Up** until the SheetName column is the first column in the list.
11. In the mapping table, insert a row for ADDRESS column in the input file that has hyperlink.
    a. Click **Insert**.
    b. In the **Excel item** option, select **Column ADDRESS**.
    c. In the **Import** option cell in the new row, select **Hyper link address**.
    d. Specify the DataStage column name EMAIL_ADDRESS for the new row.
12. Click **OK**.
13. Confirm that the values that you entered on the Configuration window are saved on the **Property** tab of the stage editor.
14. Click **Output** > **Column** tab to change the data type or other attributes. Change the type of **EMP_NO** column to `Integer`.
15. Click **OK**.

Step 3: Configuring the Sequential File stage:

Configure the Sequential File stage.

About this task

In this example, Sequential File stage is used as output stage. You can use any other output stage for creating the output.

Procedure

1. Double-click the Sequential File stage.
2. On the **Properties** page, specify the path where you want the output file to be created, followed by the file name `OutputOfExample2.txt`.
3. Click **OK**.

Step 4: Viewing the output of the job:

After you run the job, open the file, and verify the output.

Procedure

1. Save the job.
2. Compile and run the job.

   An example input Microsoft Excel files that contains the employee information for each department in the different sheets. The job extracts of employee data from all sheets are displayed in the form of following tables:

Table 9. Information of employees in DEPT_A00

EMP NO	FIRST NAME	MID INIT	LAST NAME	PHONE NO	HIRE DATE	JOB	SEX	BIRTH DATE
10	CHRISTINE	I	HAAS	3978	1/1/1995	PRES	F	8/24/1963
20	MICHAEL	L	THOMSON	3476	10/10/2003	MANAGER	M	2/2/1976
30	SALLY	A	KWAN	4738	4/5/2005	MANAGER	F	5/11/1971
50	JOHN	B	GEYER	6789	8/17/1979	MANAGER	M	9/15/1955

Table 10. Details of employees in Employees in DEPT_B01

EMP NO	FIRST NAME	MIDI NIT	LAST NAME	PHONE NO	HIRE DATE	JOB	SEX	BIRTH DATE
60	IRVING	F	STERN	6423	9/14/2003	MANAGER	M	7/7/1975
70	EVA	D	PULASKI	7831	9/30/2003	MANAGER	F	5/26/2003
90	EILEEN	W	HENDERSON	5498	8/15/2000	MANAGER	F	5/15/1971
100	THEODORE	Q	SPENSER	742	6/19/2000	MANAGER	M	12/18/1980
110	VINCENZO	G	LUCCHESSI	3490	5/16/1988	SALESREP	M	11/5/1958
120	SEAN		O'CONNELL	2167	12/51993	CLERK	M	10/18/1972
130	DELORES	M	QUINTANA	4578	7/28/2001	ANALYST	F	9/15/1955
140	HEATHER	A	NICHOLLS	1793	12/15/2006	ANALYST	F	1/19/1976
150	BRUCE		ADAMSON	4510	2/12/2002	DESIGNER	M	5/17/1972
160	ELIZABETH	R	PIANKA	3782	10/11/2006	DESIGNER	F	4/12/1980
1770	MASATOSHI	J	YOSHIMURA	2890	9/15/1999	DESIGNER	M	1/5/1981
180	MARILYN	S	SCOUTTEN	1682	7/7/2003	DESIGNER	F	2/21/1978

3. After the job runs, open the `OutputOfExample2.txt` file contains the following result.

```
"DEPT A00","10","CHRISTINE","I","HAAS","3978","1995-01-01","PRES    ",
"CHRISTINE HAAS","mailto:CHRISTINE%20HAAS@abc.com"
"DEPT A00","20","MICHAEL","L","THOMPSON","3476","2003-10-10","MANAGER ",
"MICHAEL THOMPSON","mailto:MICHAEL%20THOMPSON@abc.com"
"DEPT A00","30","SALLY","A","KWAN","4738","2005-04-05","MANAGER ",
"SALLY KWAN","mailto:SALLY%20KWAN@abc.com"
"DEPT A00","50","JOHN","B","GEYER","6789","1979-08-17","MANAGER ",
"JOHN GEYER","mailto:JOHN%20GEYER@abc.com"
"DEPT B01","60","IRVING","F","STERN","6423","2003-09-14","MANAGER ",
"IRVING STERN","mailto:IRVING%20STERN@abc.com"
"DEPT B01","70","EVA","D","PULASKI","7831","2005-09-30","MANAGER ",
"EVA PULASKI","mailto:EVA%20PULASKI@abc.com"
"DEPT B01","90","EILEEN","W","HENDERSON","5498","2000-08-15","MANAGER ",
"EILEEN HENDERSON","mailto:EILEEN%20HENDERSON@abc.com"
"DEPT B01","100","THEODORE","Q","SPENSER","972","2000-06-19","MANAGER ",
"THEODORE SPENSER","mailto:THEODORE%20SPENSER@abc.com"
"DEPT B01","110","VINCENZO","G","LUCCHESSI","3490","1988-05-16","SALESREP",
"VINCENZO LUCCHESSI","mailto:VINCENZO%20LUCCHESSI@abc.com"
"DEPT B01","120","SEAN"," ","O'CONNELL","2167","1993-12-05","CLERK   ",
"SEAN O'CONNELL","mailto:SEAN%20O'CONNELL@abc.com"
"DEPT B01","130","DELORES","M","QUINTANA","4578","2001-07-28","ANALYST ",
"DELORES QUINTANA","mailto:DELORES%20QUINTANA@abc.com"
"DEPT B01","140","HEATHER","A","NICHOLLS","1793","2006-12-15","ANALYST ",
"HEATHER NICHOLLS","mailto:HEATHER%20NICHOLLS@abc.com"
"DEPT B01","150","BRUCE"," ","ADAMSON","4510","2002-02-12","DESIGNER",
"BRUCE ADAMSON","mailto:BRUCE%20ADAMSON@abc.com"
"DEPT B01","160","ELIZABETH","R","PIANKA","3782","2006-10-11","DESIGNER",
"ELIZABETH PIANKA","mailto:ELIZABETH%20PIANKA@abc.com"
"DEPT B01","170","MASATOSHI","J","YOSHIMURA","2890","1999-09-15","DESIGNER",
"MASATOSHI YOSHIMURA","mailto:MASATOSHI%20YOSHIMURA@abc.com"
"DEPT B01","180","MARILYN","S","SCOUTTEN","1682","2003-07-07","DESIGNER",
"MARILYN SCOUTTEN","mailto:MARILYN%20SCOUTTEN@abc.com"
"DEPT C01","190","JAMES","H","WALKER","2986","2004-07-26","DESIGNER",
"JAMES WALKER","mailto:JAMES%20WALKER@abc.com"
"DEPT C01","200","DAVID"," ","BROWN","4501","2002-03-03","DESIGNER",
"DAVID BROWN","mailto:DAVID%20BROWN@abc.com"
"DEPT C01","210","WILLIAM","T","JONES","942","1998-04-11","DESIGNER",
"WILLIAM JONES","mailto:WILLIAM%20JONES@abc.com"
"DEPT C01","220","JENNIFER","K","LUTZ","672","1998-08-29","DESIGNER",
"JENNIFER LUTZ","mailto:JENNIFER%20LUTZ@abc.com"
"DEPT C01","230","JAMES","J","JEFFERSON","2094","1996-11-21","CLERK   ",
"JAMES JEFFERSON","mailto:JAMES%20JEFFERSON@abc.com"
"DEPT C01","240","SALVATORE","M","MARINO","3780","2004-12-05","CLERK   ",
"SALVATORE MARINO","mailto:SALVATORE%20MARINO@abc.com"
"DEPT C01","250","DANIEL","S","SMITH","961","1999-10-30","CLERK   ",
"DANIEL SMITH","mailto:DANIEL%20SMITH@abc.com"
"DEPT C01","260","SYBIL","P","JOHNSON","8953","2005-09-11","CLERK   ",
```

```
"SYBIL JOHNSON","mailto:SYBIL%20JOHNSON@abc.com"
"DEPT D01","270","MARIA","L","PEREZ","9001","2006-09-30","CLERK   ",
"MARIA PEREZ","mailto:MARIA%20PEREZ@abc.com"
"DEPT D01","280","ETHEL","R","SCHNEIDER","8997","1997-03-24","OPERATOR",
"ETHEL SCHNEIDER","mailto:ETHEL%20SCHNEIDER@abc.com"
"DEPT D01","290","JOHN","R","PARKER","4502","2006-05-30","OPERATOR",
"JOHN PARKER","mailto:JOHN%20PARKER@abc.com"
"DEPT D01","300","PHILIP","X","SMITH","2095","2002-06-19","OPERATOR",
"PHILIP SMITH","mailto:PHILIP%20SMITH@abc.com"
"DEPT D01","310","MAUDE","F","SETRIGHT","3332","1994-09-12","OPERATOR",
"MAUDE SETRIGHT","mailto:MAUDE%20SETRIGHT@abc.com"
"DEPT⌂D01","320","RAMLAL","V","MEHTA","9990","1995-07-07","FIELDREP",
"RAMLAL MEHTA","mailto:RAMLAL%20MEHTA@abc.com"
"DEPT D01","330","WING"," ","LEE","2103","2006-02-23","FIELDREP",
"WING LEE","mailto:WING%20LEE@abc.com"
"DEPT D01","340","JASON","R","GOUNOT","5698","1977-05-05","FIELDREP",
"JASON GOUNOT","mailto:JASON%20GOUNOT@abc.com"
```

Example 3: Extracting data from multiple ranges that have different data structures in a Microsoft Excel file:

Create a job that uses the Unstructured Data stage to extract data from multiple ranges that have different data structures in a Microsoft Excel file.

About this task

This example uses the sample Microsoft Excel file, `Employee3.xls`. This sample file has two spreadsheets, `Departments` and `Employees`, which have different data structures.

In this example, the Unstructured Data stage has two output links. you extract data from the `Departments` sheet to the first link and from the `Employees` sheet to the second link.

Step 1: Creating the job:

Create an example job that includes one Unstructured Data stage and two Sequential File stages.

Procedure

1. Start the IBM InfoSphere DataStage and QualityStage Designer client.
2. In the Repository pane, right-click the **Jobs** folder, and select **New** > **Parallel job**.
3. From the **File** section of the palette, drag an Unstructured Data stage to the canvas.
4. From the **File** section of the palette, drag two Sequential File stages to the canvas. Position the stages to the right of the Unstructured Data stage.
5. Rename the new Sequential File stages as Output_1 and Output_2.
6. Create a link from the Unstructured Data stage to the Sequential File stages.
7. Rename the links as Departments and Employees.
8. Save the job.

Step 2: Configuring the Unstructured Data stage:

Configure the Unstructured Data stage to extract data from multiple Microsoft Excel sheets.

Procedure

1. Double-click the Unstructured Data stage to open the stage properties.
2. Click **Configure**.
3. In the Configuration window, specify the full file path of the Microsoft Excel input file `Employee3.xls`.
4. Specify the data to extract from the **Departments** spreadsheet and complete the below sub steps to generate the column mappings.
   a. From the **Link** list box, select **Departments**
   b. From the **Range option** list, select **Specify the entire range**
   c. From the **Range expression** field, specify **Departments!A2:C6**
   d. From the **Column header**, select **First row of data ranges**
   e. Click **Load**. The Excel columns in the specified data range are listed in the Import pane.
5. Specify the data to extract from the **Employees** spreadsheet and complete the below sub steps to generate the column mappings.
   a. From the **Link** list box, select **Employees**.
   b. From the **Range Option** list, **Specify the entire range**
   c. From the **Range expression** field, specify **Employees!A2:L34**
   d. From the **Column header,** select **First row of data ranges**.
   e. Click **Load**. The Excel columns in the specified data range are listed in the Import pane.
   f. Click **Import**, and then click **OK**. The stage maps columns.
6. Confirm that the values that you entered on the Configuration window are saved on the **Property** tab of the stage editor.
7. On the Output page, select the **Employees** link as the Output name.
8. On the Columns page, change the data type of the EMP_NO column to integer, and then click **OK**.

Step 3: Configuring the Sequential File stages:

Configure the Sequential File stages. In this example Sequential File stage is used as output stage. You can use any other stage for creating the output.

About this task

In this example, Sequential File stage is used as output stage. You can use any other output stage for creating the output.

Procedure

1. Double-click the Sequential File stage `Output_1`.
2. On the **Properties** page, specify the path to create the output file, followed by the file name `OutputOfExample3_1.txt`.
3. Click **OK**.
4. Repeat Steps 1-3 for the second Sequential File stage `Output_2` and name file as `OutputOfExample3_2.txt`.

Step 4: Viewing the output of the job:

After you run the job, open the file, and verify the output.

Procedure

1. Save the job.
2. Compile and run the job.

   An example input Microsoft Excel file `Employee3.xls` contains department information in **Departments** sheet and employee information in **Employees** sheet. The job extracts department data to `OutputOfExample3_1.txt` file and employee data to `OutputOfExample3_2.txt`. Data in each sheet are as follows:
3. After the job runs, open the `OutputOfExample3_1.txt` file and `OutputOfExample3_2.txt` file. The `OutputOfExample3_1.txt` file should match the **Departments** sheet and `OutputOfExample3_2.txt` file should match the **Employees** sheet from the `Employee.xls` file.

Writing data to a new Microsoft Excel file

You can use the Unstructured Data stage in jobs that write data to a new Microsoft Excel file by specifying the full path name. You can also create multiple Microsoft Excel files by specifying the location from where the files are created and a prefix for the file names.

Each input link of the Unstructured Data stage is associated with a separate Microsoft Excel sheet. The Microsoft Excel sheets are named associated with the input links that are as **Sheet1**, **Sheet2**, **Sheet3**, and so on. Each InfoSphere DataStage column of the input link is mapped to a Microsoft Excel column.

Designing jobs that have the Unstructured Data stage

You can use an Unstructured Data stage to write data to unstructured data sources.

Procedure

1. Define a job that includes a Unstructured Data stage.
2. To set up the Unstructured Data stage as a target stage to write data to unstructured data sources, complete the following steps:
   a. Configure the Unstructured Data stage as a target.
   b. Specify the column definition on the link.
3. Compile and run the job.

Configuring the Unstructured Data stage as a target:

You can configure the Unstructured Data stage to generate a Microsoft Excel file.

About this task

The Unstructured Data stage supports only the OOXML (`.xlsx`) format of Microsoft Excel files as the target file.

The Unstructured Data stage supports runtime column propagation. When runtime column propagation is enabled on an output link of a upstream stage, propagated additional columns are appended after columns that are defined in the InfoSphere Designer client.

The Unstructured Data stage does not support generating `.xls` files or password-encrypted files.

Procedure

1. On the parallel canvas, double-click the **Unstructured Data** stage.
2. On the **Stage** tab, select **Excel** from the **Document type** list.
3. From the **Write mode** list, select **Create a file**.
4. Click **Configure** to configure properties for writing data to a Microsoft Excel file.
5. In the Output file pane specify the following:
   a. In the `File name` field, specify the full path name of the Microsoft Excel file to which you want to write the data. For example, you can specify `C:\tmp\employee.xlsx`.
   b. Optional: Specify the **File update** mode. If you select **Create (Error if exists)**, your job execution fails if the target Microsoft Excel file already exists. If you select **Overwrite**, which is the default setting, the Unstructured Data stage overwrites the existing file.
   c. Optional: Specify the **Write method**. If you select **Generate multiple files**, the Unstructured Data stage creates multiple Microsoft Excel files based on additional properties settings. If you select **Specific file**, which is the default setting, the Unstructured Data stage creates a Microsoft Excel file with the name that is specified in the **File name** property.
6. In the Properties pane specify the following:
   a. Optional: Clear the **Set for all links** check box to specify the properties for each input link that is selected in the **Link** list. If you select the **Set for all links** check box, which is the default setting, the properties settings are applied for all the input links.
   b. Optional: Specify the **Column header**. If you select **Column names**, the Unstructured Data stage writes InfoSphere DataStage column names to the first row of the Microsoft Excel sheet. If you select **None**, which is the default setting, the Unstructured Data stage writes the data to the first row of the Microsoft Excel sheet.
   c. Optional: Specify the **Auto size columns**. If you select **Yes**, the Unstructured Data stage adjusts each column width in the generated Microsoft Excel sheet to fit the column contents. If you select **No**, which is the default setting, the Unstructured Data stage does not adjust column width.
7. Optional: In the Sheet order pane, change the sheet order and modify the sheet names
8. Click **OK** to save the settings that you specified.

Related tasks:

"Writing data to multiple Microsoft Excel files" on page 19
You can use the Unstructured Data stage to write data to multiple Microsoft Excel files when you have a large amount of data.

Specifying the column definition on the link:

You can specify the column definition such as SQL Type, Length, Scale, and Nullable on the link.

Procedure

1. On the parallel canvas, double-click the **Unstructured Data** stage.
2. Select the **Input** tab, then select the input link from **Input name (upstream stage)**.
3. Edit the SQL type, Length, and Scale of each column.

4. Click **OK** to save the changes.

Using job parameters when modifying an existing Microsoft Excel file:

Unstructured Data stage does not have the ability to create new job parameters in Configuration window. However, you can use the job parameters in the Configuration window. You must create job parameters in the Job Properties window before or after you work on the Configuration window, by selecting **Edit** > **Job Properties** from IBM InfoSphere DataStage and QualityStage Designer client. For more information about creating job parameters, see Lesson 2.4: Adding parameters in the IBM InfoSphere DataStage Parallel Job Tutorial.

A job parameter is specified in the Configuration Window with a **#** character. For example, job parameter *FileName*, is specified as **#FileName#** in the Configuration window. For String type field such as **File name** property, you can directly type the name of job parameter within **#**.

If you want to use job parameter for the List type property such as **Action if the file already exists**, you must create a List type parameter that contains a list of string variables. The String variables must match with the label text of the corresponding property in the Configuration window. For example, if you want to use job parameter for **Action if the file already exists** property, you must create a List type job parameter that contains the string variable **Error** and **Overwrite**. After creating a job parameter, select **<Parameterize...>** from the Configuration window, and specify the job parameter name within the **#** character in the **Input Parameter** dialog box. Click **Load** to edit or select variables in the Resolve job parameters panel.

Writing data to multiple Microsoft Excel files

You can use the Unstructured Data stage to write data to multiple Microsoft Excel files when you have a large amount of data.

About this task

You can use the Unstructured Data stage to design jobs that write data to multiple Microsoft Excel files. The maximum number of records that is supported by the OOXML format of Microsoft Excel (`.xlsx`) is 1,048,576. When the input links have more than 1,048,576 records, you must divide the records into multiple Microsoft Excel files. Even if the links do not have more than 1,048,576 records, you might want to write them to multiple Microsoft Excel files because opening a large Microsoft Excel file requires a large amount of memory.

Procedure

1. Define a job that includes a Unstructured Data stage.
2. Configure the Unstructured Data stage.
3. Compile and run the job.

Related tasks:

"Configuring the Unstructured Data stage as a target" on page 17
You can configure the Unstructured Data stage to generate a Microsoft Excel file.

Configuring the Unstructured Data stage:

You can configure the Unstructured Data stage to generate a multiple Microsoft Excel files.

Procedure

1. On the parallel canvas, double-click the **Unstructured Data** stage.
2. On the **Stage** tab, select **Excel** from the **Document type** list.
3. From the **Write mode** list, select **Create a file**.
4. Click **Configure** to configure properties for writing data to a Microsoft Excel file.
5. In the Output file pane specify the following:
   a. In the **File name** field, specify the location where you want to create multiple Microsoft Excel files. Also you can specify the prefix of the file names following the file location. For example, if you want to create Microsoft Excel files at `C:\tmp` and use `Sample` as the prefix of the file names, specify `C:\tmp\Sample` in the **File name** field. The Unstructured Data stage appends a three-digit sequential number and file extension (`.xlsx`) to the prefix. If the number of files exceeds 999, the file name contains the required number of digits.
   b. Optional: Specify the **File update** mode. If you select **Create (Error if exists)**, your job execution fails if the target Microsoft Excel file already existed. If you select **Overwrite**, which is the default setting, the Unstructured Data stage overwrites the existing file.
   c. Optional: From the **Write method** list, select **Generate multiple files**. The Unstructured Data stage creates multiple Microsoft Excel files in the location that is specified in the **File name** field.
6. In the Properties pane specify the following:
   a. Optional: Select the **Set for all links** check box, which is the default setting, the properties settings are applied for all the input links.
   b. Optional: Specify the **Column header**. If you select **Column names**, the Unstructured Data stage writes InfoSphere DataStage column names to the first row of the Microsoft Excel sheet. If you select **None**, which is the default setting, the Unstructured Data stage writes the data to the first row of the Microsoft Excel sheet.
   c. Optional: Specify the **Auto size columns**. If you select **Yes**, the Unstructured Data stage adjusts each column width in the generated Microsoft Excel sheet to fit the column contents. If you select **No**, which is the default setting, the Unstructured Data stage does not adjust column width.
   d. Specify the **Maximum number of rows in a sheet**. The default is 65536. This number includes a column name row when you select **Column names** from the **Column header** list.
7. In the Sheet order pane, change the sheet order and modify the sheet names
8. Click **OK** to save the settings that you specified.

Consideration about end of wave:

In InfoSphere DataStage parallel jobs, some stages can send an end of wave marker (EOW), which indicates the end of a unit of work or transaction. When all the records that are extracted from the input link are included in a unit of work (called a single wave), the Unstructured Data stage generates Microsoft Excel sheets that contain the maximum number of records until all of the records are written.

For example, suppose that the Unstructured Data stage has two input links, DSLink1 and DSLink2. DSLink1 is associated with Sheet1 and DSLink2 is associated with Sheet2. The maximum number of records in a sheet is 65,536 and

DSLink1 has 100,000 records; DSLink2 has 150,000 records. Each sheet does not have column names in the first row. In this case, the following number of records is included in each sheet of each file.

File Name	Sheet1	Sheet2
Workbook001.xlsx	65,536	65,536
Workbook002.xlsx	34,464	65,536
Workbook003.xlsx	0	18,928

When records that are extracted from the input link are divided into two or more units of work (called multiple waves), the Unstructured Data stage stops writing records to the Microsoft Excel sheet and creates a new Microsoft Excel file if the number of records in a wave exceeds the maximum number in at least one sheet. The Unstructured Data stage does not write any records in the next wave to the previous file even if a sheet can contain more records. For example, assume that each link contains the following number of records in each wave.

Wave#	DSLink1	DSLink2
1	90,000	50,000
2	5,000	90,000
3	5,000	10,000

In the first wave, the Unstructured Data stage creates a Microsoft Excel file named `Workbook001.xlsx` that has two sheets, `Sheet1` and `Sheet2`. The Unstructured Data stage writes records that are extracted from DSLink1 to `Sheet1` until it reads the maximum number of records (65,536) and writes all the records (50,000) from DSLink2 to `Sheet2`. Next, the Unstructured Data stage creates a Microsoft Excel file named `Workbook002.xlsx` and writes the rest of the records (24,464) in the first wave and all the records (5,000) in the second wave from DSLink1 to `Sheet1`. Even though `Sheet2` of `Workbook001.xlsx` does not exceed the maximum number of records, the Unstructured Data stage writes records (65,536) in the second wave extracted from DSLink2 to `Sheet2` of `Workbook002.xlsx`, not `Workbook001.xlsx`. When the number of records in the second wave from DSLink2 exceeds the maximum number, the Unstructured Data stage creates a Microsoft Excel file named `Workbook003.xlsx` and writes the rest of the records (24,464) in the second wave from DSLink2 to `Sheet2` of `Workbook003.xlsx`. For the third wave, because both of the sheets have enough room, the Unstructured Data stage writes all of the records extracted from (5,000) and DSLink2 (10,000) to `Sheet1` and `Sheet2` of `Workbook003.xlsx`.

As a result, the following number of records is written in each sheet of each file:

File name	Sheet1	Sheet2
Workbook001.xlsx	65,536 from 1st wave	50,000 coming from 1st wave
Workbook002.xlsx	29,464 (= 24,464 from 1st wave + 5,000 from 2nd wave)	65,536 coming from 2nd wave
Workbook003.xlsx	5,000 from 3rd wave	34,464 (= 24,464 from 2nd wave + 10,000 from 3rd wave)

Examples of writing data to Microsoft Excel files

You can build sample jobs that write data to Microsoft Excel files.

To get the files for the examples, extract the `IS_install\Clients\Samples\Connectors\UnstructuredData_Samples.zip` file.

Example 1: Writing data to a Microsoft Excel spreadsheet:

Create a job that uses the Unstructured Data stage to write data to a Microsoft Excel spreadsheet

About this task

This example uses a text file, `Employee.txt` as source data. The source file contains information of employees in CSV format. You write this information to a Microsoft Excel spreadsheet.

Step 1: Creating the job:

Create an example job that includes one Unstructured Data stage and one Sequential File stage.

Procedure

1. Start the IBM InfoSphere DataStage and QualityStage Designer client.
2. In the Repository pane, right-click the **Jobs** folder, and select **New** > **Parallel job**.
3. From the **File** section of the palette drag, an Sequential File stage to the canvas.
4. From the **File** section of the palette, drag an Unstructured Data stage to the canvas. Position the stage to the right of the Sequential File stage.
5. Create a link from the Sequential File stage to the Unstructured Data stage.
6. Rename the stage and link.
7. Save the job.

Step 2: Configuring the Unstructured Data stage:

Configure the Unstructured Data stage to extract data from a range in an Microsoft Excel file.

Procedure

1. Double-click Unstructured Data stage to open the stage properties.
2. From the **Stage** tab, select **Excel** from the **Document type** list.
3. From the **Write mode** list, select **Create a file**.
4. Click **Configure**.
5. In the Configuration window, specify the full file path where you want the output file to be created, followed by the file name `OutputOfSample4.xls`.
6. From the **File update mode**, select **Overwrite**.
7. From the Write method, select Specific file.
8. From the **Column header** field, select **Column names**.
9. From the **Auto size columns** field on the Property tab, select **Yes**.
10. In Sheet order pane, specify **Employee** as Sheet name.
11. Click **OK**.

12. Confirm that the values that you specified in the Configuration window are saved on the property tab of the stage editor.
13. Click **OK**.

Step 3: Configuring the Sequential File stage:

Configure the Sequential File stage.

About this task

In this example, Sequential File stage is used as output stage. You can use any other output stage for creating the output.

Procedure

1. Double-click the Sequential File stage.
2. On the **Properties** page, specify the file path of input file `Employee.txt`.
3. On the Columns page, define the columns as shown in the below figure.

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Data element
1	EMPNO	☑	Char		6		No	
2	FIRSTNME	☐	VarChar		12		No	
3	MIDINIT	☐	Char		1		Yes	
4	LASTNAME	☐	VarChar		15		No	
5	WORKDEPT	☐	Char		3		Yes	
6	PHONENO	☐	Char		4		Yes	
7	HIREDATE	☐	Date		10		Yes	
8	JOB	☐	Char		8		Yes	
9	EDLEVEL	☐	SmallInt		2		No	
10	SEX	☐	Char		1		Yes	
11	BIRTHDATE	☐	Date		10		Yes	
12	SALARY	☐	Decimal		9	2	Yes	
13	BONUS	☐	Decimal		9	2	Yes	
14	COMM	☐	Decimal		9	2	Yes	

4. Click **OK**.

Step 4: Viewing the output of the job:

After you run the job, open the file, and verify the output.

Procedure

1. Save the job.
2. Compile and run the job.
3. View the source data in the **Employee.txt** file.

   This sample shows the beginning of the data in the file:

```
"000010","CHRISTINE","I","HAAS","A00","3978","1995-01-01","PRES","18","F","1963-08-24"," 0152750.00",
" 0001000.00"," 0004220.00"
"000020","MICHAEL","L","THOMPSON","B01","3476","2003-10-10","MANAGER ","18","M","1978-02-02",
" 0094250.00"," 0000800.00"," 0003300.00"
"000030","SALLY","A","KWAN","C01","4738","2005-04-05","MANAGER ","20","F","1971-05-11"," 0098250.00",
```

```
" 0000800.00"," 0003060.00"
"000050","JOHN","B","GEYER","E01","6789","1979-08-17","MANAGER ","16","M","1955-09-15"," 0080175.00",
" 0000800.00"," 0003214.00"
"000060","IRVING","F","STERN","D11","6423","2003-09-14","MANAGER ","16","M","1975-07-07"," 0072250.00",
" 0000500.00"," 0002580.00"
"000070","EVA","D","PULASKI","D21","7831","2005-09-30","MANAGER ","16","F","2003-05-26"," 0096170.00",
" 0000700.00"," 0002893.00"
```

4. After the job runs, open the `OutputOfExample4.xlsx` file to view the result. The following table shows the output data in the `OutputOfExample4.xlsx` file.

Table 11. Details of output data in a Microsoft Excel file

EMP NO	FIRST NAME	MID INIT	LAST NAME	PHONE NO	WORK DEPT	HIRE DATE	JOB	SEX	BIRTH DATE	SAL ARY	BON US
10	CHRIS TINE	I	HAAS	3978	A00	1/1/ 1995	PRES	F	8/24/ 1963	1527 50	1000
20	MICHAEL	L	THOM PSON	3476	A00	10/10/ 2003	MANA GER	M	2/2/1978	94250	800
30	SALLY	A	KWAN	4738	A00	4/5/ 2005	MANA GER	F	5/11/ 1971	98250	800
50	JOHN	B	GEYER	6789	A00	8/17/ 1979	MANA GER	M	9/15/ 1955	80175	800
60	IRVING	F	STERN	6423	B01	9/14/ 2003	MANA GER	M	7/7/1975	72250	500
70	EVA	D	PULASKI	7831	B01	9/30/ 2005	MANA GER	F	5/26/ 2003	96170	700
90	EILEEN	W	HEND ERSON	5498	B01	8/15/ 2000	MANA GER	F	5/15/ 1971	89750	600
100	THEO DORE	Q	SPENSER	972	B01	6/19/ 2000	MANA GER	M	12/18/ 1980	86150	500
110	VINCE NZO	G	LUCC HESSI	3490	B01	5/16/ 1988	SALES REP	M	11/5/ 1959	66500 4	900
120	SEAN		O'CON NELL	2167	B01	12/5/ 1993	CLERK	M	10/18/ 1972	9250	600
130	DELORES	M	QUIN TANA	4578	B01	7/28/ 2001	ANAL YST	F	9/15/ 1955	73800	500
140	HEATHER	A	NICHO LLS	1793	B01	12/15/ 2006	ANAL YST	F	1/19/ 1976	68420	600
150	BRUCE		ADAM SON	4510	B01	2/12/ 2002	DESIG NER	M	5/17/ 1977	55280	500
160	ELIZAB ETH	R	PIANKA	3782	B01	10/11/ 2006	DESIG NER	F	4/12/ 1980	62250	400
170	MASAT OSHI	J	YOSHI MURA	2890	B01	9/15/ 1999	DESIG NER	M	1/5/1981	44680	500
180	MARILYN	S	SCOUT TEN	1682	B01	7/7/ 2003	DESIG NER	F	2/21/ 1979	51340	500
190	JAMES	H	WALKER	2986	C01	7/26/ 2004	DESIG NER	M	6/25/ 1982	50450	400
200	DAVID		BROWN	4501	C01	3/3/ 2002	DESIG NER	M	5/29/ 1971	57740	600
210	WILLIAM	T	JONES	942	C01	4/11/ 1998	DESIG NER	M	2/23/ 2003	68270	400

Table 11. Details of output data in a Microsoft Excel file (continued)

EMP NO	FIRST NAME	MID INIT	LAST NAME	PHONE NO	WORK DEPT	HIRE DATE	JOB	SEX	BIRTH DATE	SAL ARY	BON US
220	JENNIFER	K	LUTZ	672	C01	8/29/ 1998	DESIG NER	F	3/19/ 1978	49840	600
230	JAMES	J	JEFFE RSON	2094	C01	11/21/ 1996	CLERK	M	5/30/ 1980	42180	400
240	SALVA TORE	M	MARINO	3780	C01	12/5/ 2004	CLERK	M	3/31/ 2002	48760	600
250	DANIEL	S	SMITH	961	C01	10/30/ 1999	CLERK	M	11/12/ 1969	49180	400
260	SYBIL	P	JOHN SON	8953	C01	9/11/ 2005	CLERK	F	10/5/ 1976	47250	300
270	MARIA	L	PEREZ	9001	D01	9/30/ 2006	CLERK	F	5/26/ 2003	37380	500
280	ETHEL	R	SCHNE IDER	8997	D01	3/24/ 1997	OPER ATOR	F	3/28/ 1976	36250	500
290	JOHN	R	PARKER	4502	D01	5/30/ 2006	OPER ATOR	M	7/9/1985	35340	300
300	PHILIP	X	SMITH	2095	D01	6/19/ 2002	OPER ATOR	M	10/27/ 1976	37750	400
310	MAUDE	F	SETR IGHT	3332	D01	9/12/ 1994	OPER ATOR	F	4/21/ 1961	35900	300
320	RAMLAL	V	MEHTA	9990	D01	7/7/ 1995	FIELD REP	M	8/11/ 1962	39950	400
330	WING		LEE	2103	D01	2/23/ 2006	FIELD REP	M	7/18/ 1971	45370	500
340	JASON	R	GOUNOT	5698	D01	5/5/ 1977	FIELD REP	M	5/17/ 1956	43840	500

Example 2: Writing data to multiple spreadsheets of a Microsoft Excel file:

Create a job that uses the Unstructured Data stage to write data to multiple spreadsheets of Microsoft Excel.

About this task

This example uses 7 text files, `DEPT_A00.txt`, `DEPT_B01.txt`, `DEPT_C01.txt`, `DEPT_D11.txt`, `DEPT_D21.txt`, `DEPT_E11.txt` and `DEPT_E21.txt` as source data. Each source file contains information of employees of corresponding department in CSV format. You write information from each source file to each spreadsheet. Created Microsoft Excel file has multiple spreadsheets.

Step 1: Creating the job:

Create an example job that includes seven Sequential File stages and one Unstructured Data stage.

Procedure

1. Start the IBM InfoSphere DataStage and QualityStage Designer client.

2. In the Repository pane, right-click the **Jobs** folder, and select **New** > **Parallel job**.
3. From the **File** section of the palette drag, an Sequential File stage to the canvas.
4. Repeat step 3 six more times. Position them in vertical line.
5. From the File section of the palette, drag an Unstructured Data stage to the canvas. Position the stage to the right of the Sequential File stages.
6. Create a link from a Sequential File stages to the Unstructured Data stage.
7. Rename the stages and links. Name the links so that they match the corresponding department name (A00, B01, C01, D11, D21, E11, E21).
8. Save the job.

Step 2: Configuring the Unstructured Data stage:

Configure the Unstructured Data stage to extract data from a range in an Microsoft Excel file.

Procedure

1. Double-click Unstructured Data stage to open the stage properties.
2. From the **Stage** tab, select **Excel** from the **Document type** list.
3. From the **Write mode** list, select **Create a file**.
4. Click **Configure**.
5. In the Configuration window, specify the full file path where you want the output file to be created, followed by the file name `OutputOfSample5.xls`.
   a. From the **File update mode**, select **Overwrite**.
   b. From the **Write method**, select **Specific file**. Ensure **Set for all links** option is selected.
   c. From the **Column header** field, select **Column names**.
   d. From the **Auto size columns** field on the Property tab, select **Yes**.
   e. In Sheet order pane, order links to A00, B01, C01, D11, D21, E11, E21 by using **Up** and **Down** buttons.
   f. In Sheet order pane, specify sheet names so that they match the corresponding link names.
   g. Click **OK**.
6. Click **OK**.

Step 3: Configuring the Sequential File stages:

Configure the Sequential File stages.

About this task

In this example, Sequential File stage is used as output stage. You can use any other output stage for creating the output.

Procedure

1. Double-click the Sequential File stage.
2. On the **Properties** page, specify the file path to the`DEPT_A00.txt` file.
3. On the Columns page, define the columns as shown in the below figure.

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Data element
1	EMPNO	☑	Char		6		No	
2	FIRSTNME	☐	VarChar		12		No	
3	MIDINIT	☐	Char		1		Yes	
4	LASTNAME	☐	VarChar		15		No	
5	WORKDEPT	☐	Char		3		Yes	
6	PHONENO	☐	Char		4		Yes	
7	HIREDATE	☐	Date		10		Yes	
8	JOB	☐	Char		8		Yes	
9	EDLEVEL	☐	SmallInt		2		No	
10	SEX	☐	Char		1		Yes	
11	BIRTHDATE	☐	Date		10		Yes	
12	SALARY	☐	Decimal		9	2	Yes	
13	BONUS	☐	Decimal		9	2	Yes	
14	COMM	☐	Decimal		9	2	Yes	

4. Click **OK**.
5. Repeat Step 1 to 4 six more times for the remaining source files, `DEPT_B01.txt`, `DEPT_C01.txt`, `DEPT_D11.txt`, `DEPT_D21.txt`, `DEPT_E11.txt`, and `DEPT_E21.txt`. All Sequential File stages have the same column definition on their output link.

Step 4: Viewing the output of the job:

After you run the job, open the file, and verify the output.

Procedure

1. Save the job.
2. Compile and run the job.
3. After the job runs, open the `OutputOfExample5.xlsx` file to view the result. The output data in a Microsoft Excel sheet should match the information that is in the source text file. The Microsoft Excel file `Sheet A00` should match the source data in `DEPT_A00.txt` file. The Microsoft Excel file `Sheet B01` should match the source data in `DEPT_B01.txt` file. The Microsoft Excel file `Sheet C01` should match the source data in `DEPT_C01.txt` file. The Microsoft Excel file `Sheet D11` should match the source data in `DEPT_D11.txt` file. The Microsoft Excel file `Sheet D21` should match the source data in `DEPT_D21.txt` file. The Microsoft Excel file `Sheet E11` should match the source data in `DEPT_E11.txt` file. The Microsoft Excel file `Sheet E21` should match the source data in `DEPT_E21.txt` file.

Example 3: Writing data to multiple Microsoft Excel files:

Create a job that uses the Unstructured Data stage to write data into multiple Microsoft Excel files.

About this task

This example uses a text file, `Employee.txt` as source data. The source file contains information of 42 employees in CSV format. You write this information to multiple Microsoft Excel files divided by specified maximum number of rows in a sheet option.

Step 1: Creating the job:

Create an example job that includes one Sequential File stages and one Unstructured Data stage.

Procedure

1. Start the IBM InfoSphere DataStage and QualityStage Designer client.
2. In the Repository pane, right-click the **Jobs** folder, and select **New** > **Parallel job**.
3. From the **File** section of the palette, drag a Sequential File stage to the canvas.
4. From the **File** section of the palette drag, an Unstructured Data stage to the canvas. Position the stage to the right of the Sequential File stag.
5. Create a link from the Sequential File stage to the Unstructured Data stage.
6. Rename the stage and the link.
7. Save the job.

Step 2: Configuring the Unstructured Data stage:

Configure the Unstructured Data stage to write the data to a Microsoft Excel sheet.

Procedure

1. Double-click Unstructured Data stage to open the stage properties.
2. On the **Stage** tab, select **Excel** from the **Document type** list.
3. From the **Write mode** list, select **Create a file**.
4. Click **Configure**.
5. In the Configuration window, specify the path where you want the output file to be created, followed by the file prefix `OutputOfSample6_`. When files are generated, three digits sequential number and extension `.xlsx` (e.g. 001.xlsx, 002.xlsx) are added to this prefix. For example, `001.xlsx`, `002.xlsx`.
6. From the **File update mode**, select **Overwrite**.
7. From the **Write method**, select **Generate multiple files**.
8. From the **Column header** field, select **Column names**.
9. From the **Auto size columns** field on the Property tab, select **Yes**.
10. In **Maximum number of rows in a sheet**, specify **10**
11. In Sheet order pane, specify **Employee** as Sheet name
12. Click **OK**.

Step 3: Configuring the Sequential File stages:

Configure the Sequential File stages.

About this task

In this example, Sequential File stage is used as output stage. You can use any other output stage for creating the output.

Procedure

1. Double-click the Sequential File stage.
2. On the **Properties** page, specify the file path to the `Employee.txt` file.
3. On the Columns page, define the columns as shown in the below figure.

	Column name	Key	SQL type	Extended	Length	Scale	Nullable	Data element
1	EMPNO	☑	Char		6		No	
2	FIRSTNME	☐	VarChar		12		No	
3	MIDINIT	☐	Char		1		Yes	
4	LASTNAME	☐	VarChar		15		No	
5	WORKDEPT	☐	Char		3		Yes	
6	PHONENO	☐	Char		4		Yes	
7	HIREDATE	☐	Date		10		Yes	
8	JOB	☐	Char		8		Yes	
9	EDLEVEL	☐	SmallInt		2		No	
10	SEX	☐	Char		1		Yes	
11	BIRTHDATE	☐	Date		10		Yes	
12	SALARY	☐	Decimal		9	2	Yes	
13	BONUS	☐	Decimal		9	2	Yes	
14	COMM	☐	Decimal		9	2	Yes	

4. Click **OK**.

Step 4: Viewing the output of the job:

After you run the job, open the file, and verify the output.

Procedure

1. Save the job.
2. Compile and run the job.
3. After the job runs, open the output file to view the result. The output data in a Microsoft Excel sheet should match the information that is in the source text file. Since the specified maximum number of rows in a sheet is 10 and it includes 1 row for column header, each output file includes maximum 9 records from the input file. The input file has 42 records and hence 5 files are generated. The `OutputOfExample6_001.xlsx`, `OutputOfExample6_002.xlsx`, `OutputOfExample6_003.xlsx`, `OutputOfExample6_004.xlsx`, `OutputOfExample6_005.xlsx` files are generated.

Writing data to existing Microsoft Excel files

You can use the Unstructured Data stage in jobs that write data to an existing Microsoft Excel file. You can also copy a Microsoft Excel file and write data to the copy.

When the Unstructured Data stage writes data to an existing Microsoft Excel file, the stage writes only the cell data. If a Microsoft Excel cell that the stage writes to has any format, then the stage keeps the existing format. If the Microsoft Excel file has a formula or a graph that refers to cells that are written to by the Unstructured Data stage, then the formula or the graph is recalculated when it is opened by Microsoft Excel.

You can write to Microsoft Excel columns from any InfoSphere DataStage columns. The names and order of InfoSphere DataStage columns and Microsoft Excel columns do not have to match. If the Microsoft Excel sheet has a header in the first row, you can configure the Unstructured Data stage so that values in the first row are used to determine the column that records are written to. You can write up to

1,048,576 rows of a Microsoft Excel sheet. The source InfoSphere DataStage columns must be defined in the design time.

Related reference:

"Data type conversions from InfoSphere DataStage to Microsoft Excel" on page 41
Before the Unstructured Data stage writes data that is extracted from input links to Microsoft Excel files, the data is converted to Microsoft Excel data types.

Designing jobs that write data to an existing Microsoft Excel sheet

You can use an Unstructured Data stage to design jobs that write data to an existing Microsoft Excel sheet. One Microsoft Excel file can be updated by one Unstructured Data stage at a time. You can have only one Unstructured Data stage that update the same Microsoft Excel file in one job.

Before you begin

- Install InfoSphere Information Server with the language that matches the language of the Microsoft Excel file that you want to extract.
- Ensure that the application that shows the content of Microsoft Excel spreadsheets (for example, Microsoft Excel, Microsoft Excel Viewer or IBM Lotus Symphony) is installed on your client computer.
- Ensure that the file extension `.xlsx` is associated with the application that you use to view Microsoft Excel spreadsheets.

Procedure

1. Define a job that includes a Unstructured Data stage.
2. Configure the Unstructured Data stage as a target.
3. compile and run the job.

Defining a job that includes a Unstructured Data stage:

Before you can read or write data from or to a Microsoft Excel files, you must create a job that includes the Unstructured Data stage, add any required additional stages, and create the necessary links.

Procedure

1. From the Designer client, click **File** > **New**.
2. In the New window, click the **Parallel Job** icon, and then click **OK**.
3. From the Palette, click **File**.
4. Drag the **Unstructured Data stage** icon to the canvas.
5. Create stages for the job.
6. On the left side of the Designer client in the Palette menu, select the **General** category, and then create the necessary links for the job.
7. (Optional) Double click the **Unstructured Data stage** icon to enter or modify the following attributes:
   - **Stage :** Modify the default name of the **Stage**. You can enter up to 255 characters. Alternatively, you can modify the name of the stage in the job design canvas.
   - **Description:** Enter an description of the stage.
8. Click **Save**.

Configuring the Unstructured Data stage as a target:

You can configure the Unstructured Data stage to modify an existing Microsoft Excel file. Parallel mode execution is not supported when modifying an existing Microsoft Excel file. You must configure the Unstructured Data stage in an sequential mode.

About this task

The Unstructured Data stage supports only the OOXML (`.xlsx`) format of Microsoft Excel files as the target file and template file.

The Unstructured Data stage does not support modifying `.xls` files or password encrypted files.

The Unstructured Data stage does not support Microsoft Excel files that are created by Microsoft Excel for Mac.

The file names, sheet names, and header names must be able to be expressed in the default code page of Microsoft Windows where InfoSphere DataStage Designer client is installed. If Microsoft Excel file that you want to process includes other characters then you must edit the Microsoft Excel file so that it does not include those characters.

Procedure

1. On the parallel canvas, double-click the **Unstructured Data** stage.
2. On the **Input** tab, select the input link from the **Input name (upstream stage)** field.
3. On the Columns page ensure that columns are properly defined.
4. On the **Stage** tab, select the document type as **Excel** from the Properties page.
5. From the**Write mode** list, select **Modify existing file**.
6. To configure properties for writing data to an existing Microsoft Excel file, click **Configure**.
7. Specify the target file details to write the data to.
   a. In the **File name** field, specify the name of the file to write data to.
   b. If more than 32 columns will be updated in the Microsoft Excel sheet, specify the number of the column in the **Number of columns to load** field.
   c. To load the columns in the Import pane, click **Load**.
8. From the **Link** list , select an input link to configure.
9. Specify the Microsoft Excel details to import in the Import pane.
   a. From the Sheet list, select the Microsoft Excel sheet to update.
   b. From the list of columns, select the Microsoft Excel columns to update.
   c. Click **Import**.
10. Map the imported Microsoft Excel columns to the InfoSphere DataStage columns that are defined in the input link. In the **InfoSphere DataStage column**, select an InfoSphere DataStage column to write the data.
11. Repeat Step 7 to 9 for all the input links.
12. Click **OK**.

Using job parameters when modifying an existing Microsoft Excel file:

Unstructured Data stage does not have the ability to create new job parameters in Configuration window. However, you can use the job parameters in the Configuration window. You must create job parameters in the Job Properties window before or after you work on the Configuration window, by selecting **Edit** > **Job Properties** from IBM InfoSphere DataStage and QualityStage Designer client. For more information about creating job parameters, see Lesson 2.4: Adding parameters in the IBM InfoSphere DataStage Parallel Job Tutorial.

A job parameter is specified in the Configuration Window with a **#** character. For example, job parameter *FileName*, is specified as **#FileName#** in the Configuration window. For String type field such as **File name** property, you can directly type the name of job parameter within **#**.

If you want to use job parameter for the List type property such as **Action if the file already exists**, you must create a List type parameter that contains a list of string variables. The String variables must match with the label text of the corresponding property in the Configuration window. For example, if you want to use job parameter for **Action if the file already exists** property, you must create a List type job parameter that contains the string variable **Error** and **Overwrite**. After creating a job parameter, select **<Parameterize...>** from the Configuration window, and specify the job parameter name within the **#** character in the **Input Parameter** dialog box. Click **Load** to edit or select variables in the Resolve job parameters panel.

Example: Writing data to existing Microsoft Excel files

Create a job that uses the Unstructured Data stage to write data to existing Microsoft Excel files.

About this task

This example uses a text file, `Employee.txt` as source data. The source file contains information of employees in CSV format. You write this information to Microsoft Excel file `ExcelModifySample1.xlsx`. `ExcelModifySample1.xlsx` has a sheet named Employee that contains, **EMP NO**, **FIRST NAME**, **MIDINIT**, **LAST NAME**, **HIRE DATE**, **JOB**, **SEX**, **SALARY**, **BONUS**, and **TOTAL PAY**columns. In this job, you write to these columns except **TOTAL PAY**. **TOTAL PAY** column has a formula to calculate salary and bonus.

You can build sample jobs that write data to an existing Microsoft Excel files.

To get the files for the examples, extract the `IS_install\Clients\Samples\Connectors\UnstructuredData_Samples.zip` file.

Step 1: Creating the job:

Create an example job that includes one Sequential File stage and one Unstructured Data stage. In this example, the Sequential File stage reads data from a Microsoft Excel file and then the Unstructured Data stage writes data to the Unstructured data source.

Procedure

1. Start the IBM InfoSphere DataStage and QualityStage Designer client.
2. In the Repository pane, right-click the **Jobs** folder, and select **New** > **Parallel job**.

3. From the **File** section of the palette, drag a Sequential File stage to the canvas.
4. From the **File** section of the palette drag, an Unstructured Data stage to the canvas. Position the stage to the right of the Sequential File stage.
5. Create a link from the Sequential File stage to the Unstructured Data stage.
6. Rename the stage and the link.
7. Save the job.

Step 2: Configuring the Sequential File stage:

Configure the Sequential File stage to read data from the source file. You must specify the source file name and define the column names and SQL properties.

Before you begin

In this example, Sequential File stage is used as output stage. You can use any other output stage for creating the output.

Procedure

1. Double-click the Sequential File stage.
2. On the **Properties** page, specify the file path to the `Employee.txt` file.
3. On the Columns page, define the columns as shown in the below table.

Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
EMP_NO		Integer				Yes	
FIRST_NAME		VarChar				Yes	
MIDINIT		VarChar				Yes	
LAST_NAME		VarChar				Yes	
HIRE_DATE		Date				Yes	
JOB		VarChar				Yes	
SEX		VarChar				Yes	
SALARY		Integer				Yes	
BONUS		Integer				Yes	

4. Click **OK**.

Step 3: Configuring the Unstructured Data stage:

Configure the Unstructured Data stage to write the data to existing Microsoft Excel file.

Procedure

1. Double-click Unstructured Data stage to open the stage properties.
2. From the **Stage** tab, select **Excel** from the **Document type** list.
3. From the **Write mode** list, select **Create a file**.
4. Click **Configure**.
5. In the Configuration window, specify the path where you want the output file to be created, followed by the file prefix `ExcelModifySample1`.
6. Click **Load**.
7. From the **Column header** field, select **First row**.

8. In the **Start writing from this row**, specify **2**.
9. From the **Sheet** list, select **Employee**. Ensure that Microsoft Excel columns A to I are selected.
10. Click **Import**.
11. In the Map panel, define the mapping between Microsoft Excel column and InfoSphere DataStage column.
12. Click **OK**.

Step 4: Viewing the output of the job:

After you run the job, open the file, and verify the output.

Procedure

1. Save the job.
2. Compile and run the job.
3. After the job runs, open the output file to view the result. The output data in a Microsoft Excel sheet should match the information that is in the source text file.

Chapter 2. Reference

These topics describe supported Microsoft Excel types and Microsoft Excel-type-to-DataStage-type mappings and describe job abort conditionsIBM Software Support.

Data type conversions from Microsoft Excel to InfoSphere DataStage

Before the Unstructured Data stage writes data that is extracted from Microsoft Excel to the output link, the data is converted to InfoSphere DataStage data types.

The following table shows the mapping between Microsoft Excel data types and InfoSphere DataStage data types.

Note: The Unstructured Data stage maps the data type conversions from Microsoft Excel to InfoSphere DataStage only when the Unstructured Data stage reads records from the Microsoft Excel data source.

Table 12. Mapping between Microsoft Excel cell value data types InfoSphere DataStage data types

Microsoft Excel cell data type	DataStage data type
Blank	**Integer data types** BigInt Integer SmallInt TinyInt **Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar **Fraction data types** Double Float Real **Decimal data types** Decimal Numeric **Date and time data type** Date Time Timestamp
Boolean	**Integer data types** BigInt Integer SmallInt TinyInt **Note:** Maps TRUE: 1, FALSE: 0 **Text data types** Char VarChar LongVarChar **Note:** Maps TRUE: "true", FALSE: "false" **National language text data types** NChar NVarChar LongNVarChar **Note:** Maps TRUE: "true", FALSE: "false"

Table 12. Mapping between Microsoft Excel cell value data types InfoSphere DataStage data types (continued)

Microsoft Excel cell data type	DataStage data type
Error	**Text data types** Char VarChar LongVarChar **Note:** String expression of the error. For example, #NAME? **National language text data types** NChar NVarChar LongNVarChar **Note:** String expression of the error. For example, #NAME?
Numeric	**Integer data types** BigInt Integer SmallInt TinyInt **Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar **Fraction data types** Double Float Real **Decimal data types** Decimal Numeric **Date and time data type** Date Time Timestamp
String	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar **Date and time data type** Date Time Timestamp

Table 13. Microsoft Excel other cell information data types and InfoSphere DataStage data types

Microsoft Excel other cell information data types	InfoSphere DataStage data types
Formula	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Comment	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Author of comment	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar

Table 13. Microsoft Excel other cell information data types and InfoSphere DataStage data types (continued)

Microsoft Excel other cell information data types	InfoSphere DataStage data types
Hyperlink type	**Integer data types** BigInt Integer SmallInt TinyInt **Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Hyperlink address	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Hyperlink label	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar

Table 14. Mapping between Microsoft Excel cell value data types and InfoSphere DataStage data types

Microsoft Excel cell value data types	InfoSphere DataStage data types
File name	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
File Path	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
File Size	**Integer data types** BigInt Integer SmallInt TinyInt **Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Last Modified Date	**Text data types** Char VarChar LongVarChar **Note:** String expression in yyyy-mm-dd format **National language text data types** NChar NVarChar LongNVarChar **Note:** String expression in yyyy-mm-dd format **Date and time data type** Date Time Timestamp

Table 15. Mapping between Microsoft Excel document properties and InfoSphere DataStage data types

Microsoft Excel document properties	InfoSphere DataStage data types
Authors	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Document Comments	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Content Creation Date	**Text data types** Char VarChar LongVarChar **Note:** String expression in yyyy-mm-dd format **National language text data types** NChar NVarChar LongNVarChar **Note:** String expression in yyyy-mm-dd format **Date and time data type** Date Time Timestamp
Key Words	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Revision Number	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Subject	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Title	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Company	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Category	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar

Table 15. Mapping between Microsoft Excel document properties and InfoSphere DataStage data types (continued)

Microsoft Excel document properties	InfoSphere DataStage data types
Manager	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar

Table 16. Mapping between Microsoft Excel custom property and InfoSphere DataStage data types

Mapping between Microsoft Excel custom property	InfoSphere DataStage data types
Text	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Date	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar **Date and time data type** Date Time Timestamp
Number	**Integer data types** BigInt Integer SmallInt TinyInt **Note:** If the value is an integer. **Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar **Fraction data types** Double Float Real **Decimal data types** Decimal Numeric
Boolean	**Integer data types** BigInt Integer SmallInt TinyInt **Note:** Maps TRUE: 1, FALSE: 0 **Text data types** Char VarChar LongVarChar **Note:** Maps TRUE: "true", FALSE: "false" **National language text data types** NChar NVarChar LongNVarChar **Note:** Maps TRUE: "true", FALSE: "false"

Table 17. Mapping Microsoft Excel sheet information with InfoSphere DataStage data types

Microsoft Excel sheet information	InfoSphere DataStage data types
Sheet Name	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar
Header	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar **Note:** For both text data types and National language text data types, Microsoft Excel supports special commands represented by single letter with a leading ampersand "&" in Microsoft Excel header and footer. The Unstructured Data stage does not convert those letters and just preserve them in the extracted text. Refer to `http://msdn.microsoft.com/en-us/library/dd773041%28v=office.12%29.aspx` for more information about the special commands.
Footer	**Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar **Note:** For both text data types and National language text data types, Microsoft Excel supports special commands represented by single letter with a leading ampersand "&" in Microsoft Excel header and footer. The Unstructured Data stage does not convert those letters and just preserve them in the extracted text. Refer to `http://msdn.microsoft.com/en-us/library/dd773041%28v=office.12%29.aspx` for more information about the special commands.

Table 18. Mapping between Microsoft Excel row information and their equivalent InfoSphere DataStage data types

Microsoft Excel row information	InfoSphere DataStage Data types
Row Number	**Integer data types** BigInt Integer SmallInt TinyInt **Text data types** Char VarChar LongVarChar **National language text data types** NChar NVarChar LongNVarChar

Table 18. Mapping between Microsoft Excel row information and their equivalent InfoSphere DataStage data types (continued)

Microsoft Excel row information	InfoSphere DataStage Data types
Is Hidden	**Integer data types** BigInt Integer SmallInt TinyInt **Note:** Maps TRUE: 1, FALSE: 0 **Text data types** Char VarChar LongVarChar **Note:** Maps TRUE: "true", FALSE: "false" **National language text data types** NChar NVarChar LongNVarChar **Note:** Maps TRUE: "true", FALSE: "false"

Data type conversions from InfoSphere DataStage to Microsoft Excel

Before the Unstructured Data stage writes data that is extracted from input links to Microsoft Excel files, the data is converted to Microsoft Excel data types.

The following table shows the mapping between InfoSphere DataStage data types and Microsoft Excel data types .

Note: The Unstructured Data stage maps the data type conversions from InfoSphere DataStage data types to Microsoft Excel files only when the Unstructured Data stage writes records to the Microsoft Excel data source.

Table 19. Mapping between Microsoft Excel cell value data types and InfoSphere DataStage data types

InfoSphere DataStage data type	Microsoft Excel cell data type
Text data types Char, VarChar, LongVarChar **National language text data types** NChar, NVarChar, LongNVarChar	String
Integer data types BigInt, Integer, SmallInt, TinyInt **Fraction data types** Double, Float, Real **Decimal data types** Decimal, Numeric	Numeric
Date and time data types Date, Time, Timestamp	Numeric **Note:** This note is applicable only when you are writing data to existing Microsoft Excel files. • Unstructured Data stage does not set format for date and time data types. It is recommended to set proper format to express the date, time or timestamp to your Microsoft Excel file.

Table 19. Mapping between Microsoft Excel cell value data types and InfoSphere DataStage data types (continued)

InfoSphere DataStage data type	Microsoft Excel cell data type
Binary data types Binary, VarBinary, LongVarBinary, Bit	Not supported.

Related concepts:

"Writing data to existing Microsoft Excel files" on page 29
You can use the Unstructured Data stage in jobs that write data to an existing Microsoft Excel file. You can also copy a Microsoft Excel file and write data to the copy.

Job abort conditions in Microsoft Excel

The tables describe the different job abort conditions in Microsoft Excel files.

Note: The following condition applies only when the Unstructured Data stage reads records from the Microsoft Excel files.

File name (wildcard character is not used)

When file name is used without wildcard character, the following job abort conditions can occur:

Table 20. File name (wildcard is not used)

Condition	Result
The specified file does not exist.	Job aborts.
User does not have permission to read the specified file.	Job aborts.
The specified file is not a valid Microsoft Excel file.	Job aborts.
The file cannot be opened by specified password.	Job aborts.

File name (wildcard character is used)

When file name is used with wildcard character, the following job abort conditions can occur:

Table 21. File name (wildcard is used)

Condition	Result
There is no file with the specified name.	The job continues with a warning message (no output row).
User does not have permission to read a matched file.	The job continues with a warning message if the **Error action** property is set to `Skip`. Otherwise, the job aborts.
The matched file is not a valid Microsoft Excel file.	The job continues with a warning message if the **Error action** property is set to `Skip`. Otherwise, the job aborts.
The matched file cannot be opened by the specified password.	The job continues with a warning message if the **Error action** property is set to`Skip`. Otherwise, the job aborts.

Sheet Name

The following job abort conditions can occur for sheet name:

Table 22. Sheet name

Condition	Result
Sheet name is specified in data range, and the sheet does not exist.	The job continues with a warning message if the **Error action** property is set to `Skip`. Otherwise, the job aborts.

Column header

The following job abort condition can occur for column header:

Table 23. Column header

Condition	Result
First row is column header and the value of first row does not match the value of the first row of the template data range.	The job continues and an informational message is logged.

Data type

The following job abort conditions can occur for data type:

Table 24. Data type

Condition	Result
The data type is not supported to extract the Microsoft Excel object type mapped to the DataStage column.	The job aborts.
The data type is supported to extract the Microsoft Excel object type mapped to the DataStage column, but does not match the instance (This happens when the Microsoft Excel object is a cell or a custom property).	The job continues with a warning message if the **Error action** property is set to `Skip`. Otherwise, the job aborts.

Custom property

The following job abort condition can occur for custom property:

Table 25. Custom property

Condition	Result
The specified property does not exist.	The job continues with a warning message if the **Error action** property is set to `Skip`. Otherwise, the job aborts.

Chapter 3. Troubleshooting

Use the information in this section to help you understand, isolate, and resolve issues with the InfoSphere DataStage Unstructured Data stage.

Messages displayed at the bottom of the configuration window are truncated

If you notice that the messages displayed at the bottom of the configuration window are truncated, then move the mouse over the message area to view the entire message.

Unable to get the expected template data area

If you click **Load** without specifying the Range expression or specify only the sheet name in the Range expression, then a list of template data area for column mapping are displayed. However, with some Microsoft Excel files, you might not see the template data area that is required by you.

To get a complete list of Range expression for the template data area, specify the complete Range expression for the expected template data area, then click **Load**. If you specify the start cell for the expected template data area, then you can get the template data area starting from the specified cell.

Warning message is displayed when modifying configuration for Range expression

The **Range expression** field is updated when you select one of the template data areas from the list box and click **Map**. You can view a relevant range expression that is associated with your selected template data area. However, you might want to change the range expression. For example, by default, the Unstructured Data stage returns the range expression information including the sheet name. However, you can modify the configuration to display the range expression without the sheet name. When you click **OK** to save the modified configuration, you get the following warning message:

```
The data source has been changed since the column mapping was created.
The changes might cause a runtime error. Do you want to save your changes?
```

If the changes are not required for column mapping, click **OK** to complete the configuration and confirm that the updated range expression is consistent with the mapping.

Timeout error

The following error might occur when you try to load a large Microsoft Excel file on the Configuration window and the operation did not complete within the time specified for CAS service:

```
Failed to process the request:
Failed to receive the response from the handler: Request Timed Out.
```

When this error is displayed, modify the timeout value specified with the **PropertyAdmin** command.

For example, to change the timeout value to 180 seconds, specify the following command on the services tier:

```
InformationServer/ASBServer/bin/PropertyAdmin -set -key cas.agent.timeout
-value 180
```

Where, *InformationServer* is the installation directory for InfoSphere Information Server.

Out of memory error when loading a large Microsoft Excel file

You might encounter the following error when you try to load a large Microsoft Excel file on the Configuration window:

```
The file {0} cannot be loaded because there is not enough memory.
The file might be too large (its size is {1} bytes).
Specify a smaller file as the template.
```

Where, {0}: indicates the file name specified by the user and {1}: indicates the size of the specified file, size.

When you load a large Microsoft Excel file, a large amount of Java heap memory is used. As a result, the Connection Access Service, which is a Java process that the Unstructured Data stage uses, might hang. If the process hangs, the request is cancelled and the error message is displayed.

To avoid the error, do one of the following:

- Specify a smaller Microsoft Excel file.
- Create a smaller file by copying the original Microsoft Excel file. Delete the rows and sheets that are not used for column mapping.

Error related to changing the heap size of ASBAgent

You might encounter the following error when you try to load a large Microsoft Excel file:

```
It was not found how to change the heap size of ASBAgent
```

If you enable the log view for the **ISF-CAS-NATIVE** category, you might see the following message is logged:

```
Warning message received from the native (C++) layer:
The file {0} cannot be loaded because there is not enough memory.
The file might be too large (its size is {1} bytes).
The JVM maximum heap size is {2}. The consumed heap size is {3}. {4}
```

Where,

- {0} indicates the file name specified by the user
- {1} indicates the size of the specified file
- {2} indicates the maximum heap size for JVM
- {3} indicates the heap size currently consumed

To workaround the issue, increase the Java heap memory size of the ASBAgent.

Java runtime exception error

You might encounter the following fatal error when a large Microsoft Excel file is being processed:

```
Unstructured_Data_0,0: Java runtime exception occurred: java.lang.OutOfMemoryError
(java.util.Arrays::copyOfRange, file Arrays.java, line 4,138)
```

The error occurs as there is not enough Java heap memory size.

To workaround, increase the available Java heap size by setting the environment variable *CC_UNST_JAVA_HEAP*. The value of the environment variable is the integer value of Java heap size in MB. For example, to set Java heap size to 512 MB, set CC_UNST_JAVA_HEAP=512. The default heap size is 256 MB.

Extract Microsoft Excel file		
	Microsoft Excel 97-2003 (.xls) file	6 times of the file size
	Password protected Microsoft Excel 2007-2010 (.xlsx) file	30 times of the file size
	Unprotected Microsoft Excel 2007-2010 (.xlsx) file	Default heap size
Modify Microsoft Excel file		300 times of the result Microsoft Excel file size
Create Microsoft Excel file		Default heap size

Note: The actual required heap size varies depending on the type of data the Microsoft Excel file contains and platform, and may exceed the size shown in the above table.

Unable to find a proper range to specify

- You might not be able to find a proper range expression to specify when you want to extract data ranges from multiple sheets or multiple files in the following situation.
  - When each data range starts from different positions
  - When end of data ranges are not the last row in the sheets

In such cases, define name of the data ranges in the source Microsoft Excel files and specify the name as range expression.

Unable to compile a job created in old version

You might encounter the following error when compiling a job that is created in InfoSphere DataStage version 9.1

Note: A value is not specified for the Range expression property. From the stage editor, click **Configure**, and then specify a value for the property. If this job was created in InfoSphere DataStage version 9.1, you need to open configuration window and save the job.

This error occurs when you do either of the following .

1. Open Stage Editor.
2. Click **OK** on Stage Editor without clicking on **Configure**.

or

1. Open Stage Editor.
2. Click **Configure**.
3. Click **Cancel** on Configuration Window.

4. Click **OK** on Stage Editor.

Note: If you do not open the Stage Editor or do not click **OK** on Stage Editor, you can run a job created on InfoSphere DataStage version 9.1 without any actions.

To fix this error, follow the below steps.

1. Open Stage Editor.
2. Click **Configure**.
3. Click **OK** on Configuration Window.
4. Click **OK** on Stage Editor.

Chapter 4. Environment variables: Unstructured Data stage

The Unstructured Data stage uses these environment variables.

CC_JNI_EXT_DIRS

Set this environment variable to add a prefix to the class path of `java.ext.dirs` system property.

When the value of this environment variable is set, a prefix is added to the class path of `java.ext.dirs` system property.

CC_JVM_OPTIONS

Set this environment variable to specify the JVM arguments that are used when a job is run.

When this variable is specified, it takes precedence over the value of the default JVM arguments for the Java-based connectors. For example, if you set **`CC_JVM_OPTIONS`** as `-Xmx512M`, the maximum heap size is set to 512 MB when JVM instances for the connector are created.

CC_JVM_OVERRIDE_OPTIONS

Set this environment variable to override the JVM options for the conductor node so that you can avoid or fix a possible conflict.

In the conductor node in a parallel job, Java connectors are initialized for schema reconciliation. Therefore, all Java connectors in a job are initialized in the same JVM. In a single job, multiple stages might be developed in Java. Each of these stages might define JVM options such as class path, system property, heap size and so on. If two stages are run in the same physical JVM, the JVM options might conflict with each other.

CC_IGNORE_TIME_LENGTH_AND_SCALE

Set this environment variable to change the behavior of the connector on the parallel canvas.

When this environment variable is set to 1, the connector running with the parallel engine ignores the specified length and scale for the timestamp column. For example, when the value of this environment variable is not set and if the length of the timestamp column is 26 and the scale is 6, the connector on the parallel canvas considers that the timestamp has a microsecond resolution. When the value of this environment variable is set to 1, the connector on the parallel canvas does not consider that the timestamp has a microsecond resolution unless the microseconds extended property is set even if the length of the timestamp column is 26 and the scale is 6.

CC_MSG_LEVEL

Set this environment variable to specify the minimum severity of the messages that the connector reports in the log file.

At the default value of 3, informational messages and messages of a higher severity are reported to the log file.

The following list contains the valid values:

- 1 - Trace
- 2 - Debug
- 3 - Informational
- 4 - Warning
- 5 - Error
- 6 - Fatal

CC_UNST_JAVA_HEAP

Set this environment variable to control the size of the Java heap that can be used by the Unstructured Data stage.

Set the variable to an integer value that represents the Java heap size in MB. For example, to set the Java heap size to 512 MB, set **CC_UNST_JAVA_HEAP** to 512. The default Java heap size is 256 MB.

Appendix A. Product accessibility

You can get information about the accessibility status of IBM products.

The IBM InfoSphere Information Server product modules and user interfaces are not fully accessible.

For information about the accessibility status of IBM products, see the IBM product accessibility information at http://www.ibm.com/able/product_accessibility/index.html.

Accessible documentation

Accessible documentation for products is provided in IBM Knowledge Center. IBM Knowledge Center presents the documentation in XHTML 1.0 format, which is viewable in most web browsers. Because IBM Knowledge Center uses XHTML, you can set display preferences in your browser. This also allows you to use screen readers and other assistive technologies to access the documentation.

The documentation that is in IBM Knowledge Center is also provided in PDF files, which are not fully accessible.

IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility.

Appendix B. Reading command-line syntax

This documentation uses special characters to define the command-line syntax.

The following special characters define the command-line syntax:

[ ] Identifies an optional argument. Arguments that are not enclosed in brackets are required.

... Indicates that you can specify multiple values for the previous argument.

| Indicates mutually exclusive information. You can use the argument to the left of the separator or the argument to the right of the separator. You cannot use both arguments in a single use of the command.

{ } Delimits a set of mutually exclusive arguments when one of the arguments is required. If the arguments are optional, they are enclosed in brackets ([ ]).

Note:

- The maximum number of characters in an argument is 256.
- Enclose argument values that have embedded spaces with either single or double quotation marks.

For example:

wsetsrc*[-S server] [-l label] [-n name] source*

The *source* argument is the only required argument for the **wsetsrc** command. The brackets around the other arguments indicate that these arguments are optional.

wlsac *[-l | -f format] [key...] profile*

In this example, the -l and -f format arguments are mutually exclusive and optional. The *profile* argument is required. The *key* argument is optional. The ellipsis (...) that follows the *key* argument indicates that you can specify multiple key names.

wrb -import *{rule_pack | rule_set}...*

In this example, the rule_pack and rule_set arguments are mutually exclusive, but one of the arguments must be specified. Also, the ellipsis marks (...) indicate that you can specify multiple rule packs or rule sets.

Appendix C. How to read syntax diagrams

The following rules apply to the syntax diagrams that are used in this information:

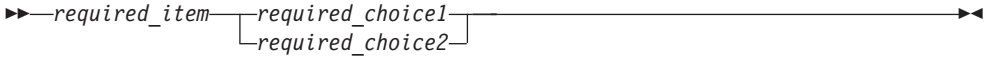
- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
  - The >>--- symbol indicates the beginning of a syntax diagram.
  - The ---> symbol indicates that the syntax diagram is continued on the next line.
  - The >--- symbol indicates that a syntax diagram is continued from the previous line.
  - The --->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).

- Optional items appear below the main path.

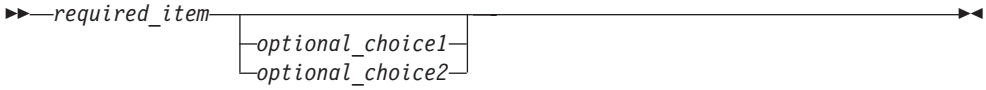
  If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.

- If you can choose from two or more items, they appear vertically, in a stack.

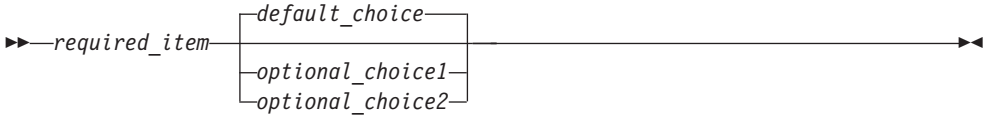
  If you must choose one of the items, one item of the stack appears on the main path.



  If choosing one of the items is optional, the entire stack appears below the main path.

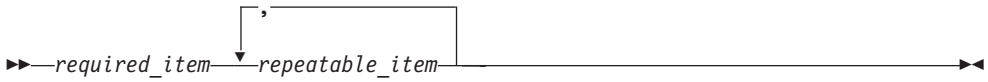


  If one of the items is the default, it appears above the main path, and the remaining choices are shown below.



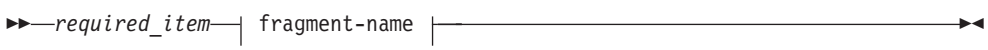
- An arrow returning to the left, above the main line, indicates an item that can be repeated.

If the repeat arrow contains a comma, you must separate repeated items with a comma.



A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Sometimes a diagram must be split into fragments. The syntax fragment is shown separately from the main syntax diagram, but the contents of the fragment should be read as if they are on the main path of the diagram.



Fragment-name:

- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown.
- Variables appear in all lowercase italic letters (for example, **column-name**). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols, exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses, for example (1).

Appendix D. Contacting IBM

You can contact IBM for customer support, software services, product information, and general information. You also can provide feedback to IBM about products and documentation.

The following table lists resources for customer support, software services, training, and product and solutions information.

Table 26. IBM resources

Resource	Description and location
IBM Support Portal	You can customize support information by choosing the products and the topics that interest you at www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server
Software services	You can find information about software, IT, and business consulting services, on the solutions site at www.ibm.com/businesssolutions/
My IBM	You can manage links to IBM Web sites and information that meet your specific technical support needs by creating an account on the My IBM site at www.ibm.com/account/
Training and certification	You can learn about technical training and education services designed for individuals, companies, and public organizations to acquire, maintain, and optimize their IT skills at http://www.ibm.com/training
IBM representatives	You can contact an IBM representative to learn about solutions at www.ibm.com/connect/ibm/us/en/

Appendix E. Accessing the product documentation

Documentation is provided in a variety of formats: in the online IBM Knowledge Center, in an optional locally installed information center, and as PDF books. You can access the online or locally installed help directly from the product client interfaces.

IBM Knowledge Center is the best place to find the most up-to-date information for InfoSphere Information Server. IBM Knowledge Center contains help for most of the product interfaces, as well as complete documentation for all the product modules in the suite. You can open IBM Knowledge Center from the installed product or from a web browser.

Accessing IBM Knowledge Center

There are various ways to access the online documentation:

- Click the **Help** link in the upper right of the client interface.
- Press the F1 key. The F1 key typically opens the topic that describes the current context of the client interface.

  Note: The F1 key does not work in web clients.
- Type the address in a web browser, for example, when you are not logged in to the product.

  Enter the following address to access all versions of InfoSphere Information Server documentation:

  `http://www.ibm.com/support/knowledgecenter/SSZJPZ/`

  If you want to access a particular topic, specify the version number with the product identifier, the documentation plug-in name, and the topic path in the URL. For example, the URL for the 11.3 version of this topic is as follows. (The ⇒ symbol indicates a line continuation):

  ```
  http://www.ibm.com/support/knowledgecenter/SSZJPZ_11.3.0/⇒
  com.ibm.swg.im.iis.common.doc/common/accessingiidoc.html
  ```

  Tip:

  The knowledge center has a short URL as well:

  `http://ibm.biz/knowctr`

  To specify a short URL to a specific product page, version, or topic, use a hash character (#) between the short URL and the product identifier. For example, the short URL to all the InfoSphere Information Server documentation is the following URL:

  `http://ibm.biz/knowctr#SSZJPZ/`

  And, the short URL to the topic above to create a slightly shorter URL is the following URL (The ⇒ symbol indicates a line continuation):

  ```
  http://ibm.biz/knowctr#SSZJPZ_11.3.0/com.ibm.swg.im.iis.common.doc/⇒
  common/accessingiidoc.html
  ```

Changing help links to refer to locally installed documentation

IBM Knowledge Center contains the most up-to-date version of the documentation. However, you can install a local version of the documentation as an information center and configure your help links to point to it. A local information center is useful if your enterprise does not provide access to the internet.

Use the installation instructions that come with the information center installation package to install it on the computer of your choice. After you install and start the information center, you can use the **iisAdmin** command on the services tier computer to change the documentation location that the product F1 and help links refer to. (The ⇒ symbol indicates a line continuation):

Windows

```
IS_install_path\ASBServer\bin\iisAdmin.bat -set -key ⇒
com.ibm.iis.infocenter.url -value http://<host>:<port>/help/topic/
```

AIX® Linux

```
IS_install_path/ASBServer/bin/iisAdmin.sh -set -key ⇒
com.ibm.iis.infocenter.url -value http://<host>:<port>/help/topic/
```

Where <host> is the name of the computer where the information center is installed and <port> is the port number for the information center. The default port number is 8888. For example, on a computer named server1.example.com that uses the default port, the URL value would be http://server1.example.com:8888/help/topic/.

Obtaining PDF and hardcopy documentation

- The PDF file books are available online and can be accessed from this support document: https://www.ibm.com/support/docview.wss?uid=swg27008803&wv=1.
- You can also order IBM publications in hardcopy format online or through your local IBM representative. To order publications online, go to the IBM Publications Center at http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss.

Appendix F. Providing feedback on the product documentation

You can provide helpful feedback regarding IBM documentation.

Your feedback helps IBM to provide quality information. You can use any of the following methods to provide comments:

- To provide a comment about a topic in IBM Knowledge Center that is hosted on the IBM website, sign in and add a comment by clicking **Add Comment** button at the bottom of the topic. Comments submitted this way are viewable by the public.
- To send a comment about the topic in IBM Knowledge Center to IBM that is not viewable by anyone else, sign in and click the **Feedback** link at the bottom of IBM Knowledge Center.
- Send your comments by using the online readers' comment form at www.ibm.com/software/awdtools/rcf/.
- Send your comments by e-mail to comments@us.ibm.com. Include the name of the product, the version number of the product, and the name and part number of the information (if applicable). If you are commenting on specific text, include the location of the text (for example, a title, a table number, or a page number).

Notices and trademarks

This information was developed for products and services offered in the U.S.A. This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

Notices

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session or persistent cookies. If a product or component is not listed, that product or component does not use cookies.

Table 27. Use of cookies by InfoSphere Information Server products and components

Product module	Component or feature	Type of cookie that is used	Collect this data	Purpose of data	Disabling the cookies
Any (part of InfoSphere Information Server installation)	InfoSphere Information Server web console	• Session • Persistent	User name	• Session management • Authentication	Cannot be disabled
Any (part of InfoSphere Information Server installation)	InfoSphere Metadata Asset Manager	• Session • Persistent	No personally identifiable information	• Session management • Authentication • Enhanced user usability • Single sign-on configuration	Cannot be disabled

Table 27. Use of cookies by InfoSphere Information Server products and components (continued)

Product module	Component or feature	Type of cookie that is used	Collect this data	Purpose of data	Disabling the cookies
InfoSphere DataStage	Big Data File stage	• Session • Persistent	• User name • Digital signature • Session ID	• Session management • Authentication • Single sign-on configuration	Cannot be disabled
InfoSphere DataStage	XML stage	Session	Internal identifiers	• Session management • Authentication	Cannot be disabled
InfoSphere DataStage	IBM InfoSphere DataStage and QualityStage Operations Console	Session	No personally identifiable information	• Session management • Authentication	Cannot be disabled
InfoSphere Data Click	InfoSphere Information Server web console	• Session • Persistent	User name	• Session management • Authentication	Cannot be disabled
InfoSphere Data Quality Console		Session	No personally identifiable information	• Session management • Authentication • Single sign-on configuration	Cannot be disabled
InfoSphere QualityStage Standardization Rules Designer	InfoSphere Information Server web console	• Session • Persistent	User name	• Session management • Authentication	Cannot be disabled
InfoSphere Information Governance Catalog		• Session • Persistent	• User name • Internal identifiers • State of the tree	• Session management • Authentication • Single sign-on configuration	Cannot be disabled
InfoSphere Information Analyzer	Data Rules stage in the InfoSphere DataStage and QualityStage Designer client	Session	Session ID	Session management	Cannot be disabled

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies:

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The United States Postal Service owns the following trademarks: CASS, CASS Certified, DPV, LACS^{Link}, ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS and United States Postal Service. IBM Corporation is a non-exclusive DPV and LACS^{Link} licensee of the United States Postal Service.

Other company, product or service names may be trademarks or service marks of others.

Index

Printed in USA

SC19-4330-00