

IBM InfoSphere DataStage and QualityStage  
Version 11 Release 3

*Parallel Job Tutorial*





IBM InfoSphere DataStage and QualityStage  
Version 11 Release 3

*Parallel Job Tutorial*



**Note**

Before using this information and the product that it supports, read the information in “Notices and trademarks” on page 69.

---

# Contents

## Chapter 1. Tutorial: Creating parallel jobs . . . . . 1

## Chapter 2. Setting up the parallel job tutorial environment . . . . . 3

Creating a folder for the tutorial files . . . . .	3
Creating the project for the parallel job tutorial . . . . .	3
Copying source files for the parallel job tutorial . . . . .	4
Importing components for the parallel job tutorial . . . . .	4
Creating a database table for the parallel job tutorial . . . . .	5
Creating a DSN for the parallel job tutorial table . . . . .	5
Creating a DSN for the tutorial table on a Microsoft Windows computer. . . . .	5
Creating a DSN for the tutorial table on an AIX, HP-UX, Solaris, or Linux computer . . . . .	6

## Chapter 3. Module 1: Opening and running the sample job. . . . . 11

Lesson 1.1: Opening the sample job . . . . .	11
Procedure . . . . .	12
Lesson 1.2: Viewing and compiling the sample job . . . . .	13
Exploring the Sequential File stage . . . . .	13
Exploring the Data Set stage. . . . .	14
Compiling the sample job . . . . .	14
Lesson 1.3: Running the sample job . . . . .	15
Running the job . . . . .	15
Viewing the output. . . . .	15
What's next . . . . .	16

## Chapter 4. Module 2: Designing your first job . . . . . 17

Lesson 2.1: Creating your job . . . . .	17
Creating a new job . . . . .	18
Adding stages and linking them . . . . .	18
Lesson 2.2: Specifying metadata . . . . .	19
Specifying column metadata. . . . .	19
Specifying lookup properties and running the job . . . . .	21
Lesson 2.3: Importing metadata. . . . .	22
Importing metadata into your repository . . . . .	22
Loading column metadata from the repository. . . . .	23
Lesson 2.4: Adding parameters . . . . .	24
Defining job parameters . . . . .	24
Adding job parameters to your job . . . . .	25
Running your job with parameters . . . . .	26
Lesson 2.5: Creating a parameter set . . . . .	26
Procedure . . . . .	26
What's next . . . . .	27

## Chapter 5. Module 3: Transforming data 29

Lesson 3.1: Designing your transformation job. . . . .	29
Creating your transformation job . . . . .	30
Configuring the Data Set stages . . . . .	30
Configuring the Transformer stage. . . . .	31

Running your transformation job . . . . .	33
Lesson 3.2: Using a Lookup stage in your transformation job . . . . .	34
Creating a lookup job . . . . .	35
Configuring the Lookup File Set stage . . . . .	36
Configuring the Lookup stage . . . . .	37
Lesson 3.3: Capturing rejected data . . . . .	38
Procedure . . . . .	38
Lesson 3.4: Completing multiple transformations in a single job . . . . .	40
Adding transformation and lookup operations. . . . .	40
Adding transformation logic to your job. . . . .	41
Configuring the Lookup operation. . . . .	44
What's next . . . . .	46

## Chapter 6. Module 4: Loading data to a relational database . . . . . 47

Lesson 4.1: Creating a data connection object . . . . .	47
Overview . . . . .	47
Creating a data connection object . . . . .	48
Lesson 4.2: Importing column metadata from a database table . . . . .	48
Overview . . . . .	48
Procedure . . . . .	49
Lesson 4.3: Writing data to a database . . . . .	50
Overview . . . . .	50
Creating the job . . . . .	50
Configuring the target. . . . .	51
Configuring the ODBC connector . . . . .	51
What's next . . . . .	53

## Appendix A. Product accessibility . . . 55

## Appendix B. Contacting IBM . . . . . 57

## Appendix C. Accessing the product documentation . . . . . 59

## Appendix D. Product accessibility . . . 61

## Appendix E. Contacting IBM . . . . . 63

## Appendix F. Accessing the product documentation . . . . . 65

## Appendix G. Providing feedback on the product documentation . . . . . 67

## Notices and trademarks . . . . . 69

## Index . . . . . 75



---

## Chapter 1. Tutorial: Creating parallel jobs

In this tutorial, you use InfoSphere® DataStage® to develop jobs that extract, transform, and load data. By transforming and cleansing the source data and applying consistent formatting, you enhance the quality of the data.

In this scenario, the worldwide companies, GlobalCo and WorldCo, are merging. Because their businesses are similar, the two companies have many customers in common. The merged company, GlobalCo Worldwide, wants to build a data warehouse for their delivery and billing information. The exercises in this tutorial focus on a small portion of the work that needs to be done to accomplish this goal.

Your part of the project is to work on the GlobalCo data that includes billing records for customer data. You read this data from comma-separated values (CSV) files, and then cleanse and transform the data, in preparation for it to be merged with the equivalent data from WorldCo. This data forms the GlobalCo\_billing dimension table in the finished data warehouse. Another developer merges this dimension table with the WorldCo\_billing dimension table to create the billing information for GlobalCo Worldwide.

This tutorial guides you through the tasks that you complete to extract, transform, and load the billing data for GlobalCo. The following steps summarize the sequence of actions that you complete:

1. In Module 1, you open the sample job and explore each stage that the job is composed of. Understanding how the stages operate is important before you begin designing your own job. You also learn how to compile the job, run the job, and view the generated output.
2. In Module 2, you create your first job by adding stages and links to the InfoSphere DataStage and QualityStage® Designer canvas. You then import metadata to create table definitions that you use throughout this tutorial. You also create parameters and parameter sets that you reuse to simplify your job design and promote reuse across your jobs.
3. In Module 3, you design a transformation job that cleanses the GlobalCo billing data so that it can be merged with the WorldCo billing data. You then expand this job by adding multiple transformations to your job to add stricter data typing to the billing data.
4. In Module 4, you load the cleansed GlobalCo billing data into a relational database so that other developers in the GlobalCo Worldwide organization can access the data. You create a data connection object, import column metadata from a database table, and then write the output to an existing table in the database.

### Learning objectives

As you work through the tutorial, you learn how to complete the following tasks:

- Design parallel jobs that extract, transform, and load data
- Create reusable objects that can be included in other job designs
- Modify your job design to implement stricter data typing
- Run the jobs that you design and view the results

## Time required

Before you begin the tutorial, ensure that your InfoSphere DataStage and QualityStage Administrator completed the steps in Chapter 2, “Setting up the parallel job tutorial environment,” on page 3. The time that is required to install and configure the tutorial depends on your InfoSphere DataStage environment.

This tutorial takes approximately four hours to finish. If you explore other concepts related to this tutorial, it can take longer to complete.

## System requirements

This tutorial requires the following hardware and software:

- InfoSphere DataStage clients installed on a Windows platform
- A connection to an InfoSphere DataStage server that runs on a Windows or UNIX platform

**Tip:** Windows servers can be on the same computer as the clients

## Prerequisites

This tutorial is intended for novice InfoSphere DataStage designers who want to learn how to create parallel jobs. Knowing about basic InfoSphere DataStage concepts, such as jobs, stages, and links might be helpful, but is not required.

**Notices:** GlobalCo, WorldCo, and GlobalCo Worldwide depict fictitious business operations with sample data used to develop sample applications for IBM and IBM customers. These fictitious records include sample data for sales transactions, product distribution, finance, and human resources. Any resemblance to actual names, addresses, contact numbers, or transaction values, is coincidental. Other sample files might contain fictional data manually or machine generated, factual data compiled from academic or public sources, or data used with permission of the copyright holder, for use as sample data to develop sample applications. Product names referenced might be the trademarks of their respective owners. Unauthorized duplication is prohibited.

---

## Chapter 2. Setting up the parallel job tutorial environment

Before you can start the tutorial, your IBM® InfoSphere DataStage and QualityStage Administrator must create folders, create the tutorial project, import source files, and complete other setup tasks.

### Before you begin

You need InfoSphere DataStage and QualityStage privileges and Microsoft Windows administrator privileges to complete some of the installation and setup tasks. You need a higher level of system knowledge and database knowledge to complete the installation and setup tasks than you need to complete the tutorial.

---

### Creating a folder for the tutorial files

Create a folder on your InfoSphere DataStage client computer and copy the files from the installation CD to the folder.

#### Procedure

1. Navigate to the `\TutorialData\DataStage\parallel_tutorial` folder, which is on the installation media. The folder might be in a directory named `is-client` or `is-suite`.
2. Create a new folder on your computer (for example, `C:\tutorial`).
3. Copy the folder on the CD named `\TutorialData\DataStage\parallel_tutorial` to the folder that you created on the client computer.

---

### Creating the project for the parallel job tutorial

Create a new project for the tutorial to keep the tutorial exercises separate from other InfoSphere DataStage projects.

### Before you begin

You create the tutorial project by using the IBM InfoSphere DataStage and QualityStage Administrator client. You must have sufficient permissions for this tool to create the tutorial project.

#### Procedure

1. Select **Start > Programs > IBM InfoSphere Information Server > IBM InfoSphere DataStage and QualityStage Administrator**
2. In the Attach window, type your user name and password.
3. In the Administrator client window, click the **Projects** tab.
4. In the Projects page, click **Add**.
5. In the **Name** field, enter `Tutorial` as the name of the new project, then click **OK**.
6. Click **Close** to close the Administrator client.

---

## Copying source files for the parallel job tutorial

You downloaded source files for the tutorial to your computer, but you need these files to be accessible by the computer that hosts the engine tier. Copy the tutorial data files from the Tutorial directory on the computer that hosts the client tier to the project directory on the computer that hosts the engine tier.

### About this task

When you created the Tutorial project, a directory was generated on the computer that hosts the engine tier. The engine tier might be on the same Windows computer that hosts the client tier, or it might exist on a separate computer. Determine where the directory for the Tutorial project was created so that you can copy the source files to that directory.

### Procedure

1. Open the directory where you extracted the source files for the tutorial. This file contains three .csv files, three .ddl files, and one .dsx file.
2. Navigate to the Projects directory and open the Tutorial directory for your project. The following directories are the default Projects directories for InfoSphere DataStage.

Option	Description
Microsoft Windows	C:\IBM\InformationServer\Server\Projects\
AIX, HP-UX, Solaris, Linux	/opt/IBM/InformationServer/Server/Projects/

3. In this directory, create a new directory named Tutorial\_files.
4. Copy the contents of the Tutorial directory from the computer that hosts the client tier to the Tutorial\_files directory on the computer that hosts the engine tier.

---

## Importing components for the parallel job tutorial

Use the Designer client to import the sample job and sample table definitions.

### Procedure

1. Select **Start > Programs > IBM InfoSphere Information Server > IBM InfoSphere DataStage and QualityStage Designer**.
2. In the Attach window, type your user name and password.
3. Select the Tutorial project from the **Project** list, and then click **OK**.
4. In the New window, click **Cancel** to close. Because you are opening an existing job, you can close this window.
5. From the Designer client menu bar, click **Import > DataStage Components**.
6. In the DataStage Repository Import window, in the **Import from file** field, type *Tutorial\_files\parallel\_tutorial.dsx*. **Tutorial\_files** is the name of the folder that you created on the computer that hosts the engine tier.
7. Click the **Import all** radio button.
8. Click **OK**. The Designer client imports the sample job and sample table definitions into the directory that you selected.
9. Click **File > Exit** to close the Designer client.

---

## Creating a database table for the parallel job tutorial

Create a table in the relational database that hosts the metadata repository for InfoSphere DataStage.

### Before you begin

If you are creating a target database table on a Microsoft Windows computer, you must have Administrator privileges or be able to run commands as an Administrator.

Run the following command to start an Administrator session:

```
runas /User:Administrator cmd.exe
```

### About this task

InfoSphere DataStage uses a repository that is hosted by a relational database. Module 4 of the tutorial imports metadata from a table in a relational database and then writes data to the table. You use data definition language (DDL) scripts to create the database table.

### Procedure

1. Create a new database named Tutorial1. You must create this database in the UTF-8 code set. The database creation scripts fail if you run them on a database that is not in the UTF-8 code set.
2. Connect to the new database.
3. Navigate to the Tutorial\_files directory in the InfoSphere DataStage Projects directory.
4. Run the appropriate DDL script to create the tutorial table in the new database.

Database type	Script
IBM DB2®	DB2_table.ddl
Oracle	Oracle_table.ddl
Microsoft SQL Server	SQLserver_table.ddl

---

## Creating a DSN for the parallel job tutorial table

Create a DSN for the database table so that users of the parallel job tutorial can connect to the table by using an ODBC connection.

### Creating a DSN for the tutorial table on a Microsoft Windows computer

You define the DSN on the computer where the IBM InfoSphere Information Server engine is installed. Users of the parallel job tutorial use this DSN in the Designer client to connect to the tutorial table.

### Before you begin

You must have administrator privileges on the Windows computer to complete this task.

## Procedure

1. Open the Control Panel and select **Administrative Tools**.
2. Select **Data Sources (ODBC)**.
3. Click the **System DSN** tab.
4. In the System DSN page, click **Add**.
5. In the Create New Data Source window, select a driver for the database, and then click **Finish**.

**Important:** If your database driver is not listed, then you must install the database driver.

6. In the database driver window that opens, specify the details that the driver requires to connect to the tutorial database. If you specified a user name and password for the database, then include these connection details.
7. Close the database driver window, then click **OK** to close the ODBC Data Source Administrator window.

## Creating a DSN for the tutorial table on an AIX, HP-UX, Solaris, or Linux computer

You define the DSN on the computer where the IBM InfoSphere Information Server engine is installed. Users of the parallel job tutorial use this DSN in the Designer client to connect to the tutorial table.

### Before you begin

To set up a DSN on an AIX, HP-UX, Solaris, or Linux computer, you edit three files:

- dsenv
- odbc.ini
- uvodbc.config

The entries that you make in each file depend on your operating system and the type of database that you are connecting to.

### Configuring the dsenv file

For some ODBC connections, plug-ins, and connectors, and for interactions with external applications such as IBM WebSphere® MQ, you must add environment variables to enable interactive use of ODBC drivers to make a connection to an ODBC data source.

### Before you begin

You must be logged in as an InfoSphere DataStage administrator with the operating system credentials on the server for the InfoSphere DataStage administrator.

Back up the dsenv file before you edit it.

### About this task

The dsenv file contains a series of shell arguments that are used when the engine starts. Interactive users, other programs, and scripts can use the dsenv file. For a connection that uses a wire protocol driver, you do not have to modify the dsenv file.

## Procedure

1. Edit the dsenv file. The file is located in the \$DSHOME/DSEngine directory, where \$DSHOME identifies the InfoSphere DataStage installation directory. The default directory is /opt/IBM/InformationServer/Server/DSEngine.
2. Specify the following information in the dsenv file:
  - Environment variables for the database client software
  - Database home location
  - Database library directory

Table 1. Names of the library path environment variable, by operating system

Operating system	Library path environment variable
AIX®	LIBPATH
HP-UX	SHLIB_PATH
HP-UX on Intel Itanium	LD_LIBRARY_PATH
Linux	LD_LIBRARY_PATH
Solaris	LD_LIBRARY_PATH

The following examples show typical entries for commonly used databases. The entries vary slightly depending on your operating system. See the data source documentation for more information.

### Sybase 11

```
LANG=
export LANG
SYBASE=/export/home/sybase/sybase
export SYBASE
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$SYBASE/lib:/usr/lib:/lib
export LD_LIBRARY_PATH
```

### Informix® XPS 9.3

```
INFORMIXDIR=/space/informix
export INFORMIXDIR
INFORMIXSERVER=solnew2.1
export INFORMIXSERVER
ONCONFIG=onconfig.solnew2
export ONCONFIG
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INFORMIXDIR/lib:$INFORMIXDIR/lib/
esql:$INFORMIXDIR/lib/cli
export LD_LIBRARY_PATH
LANG=C
export LANG
```

### DB2 Version 10.5

```
DB2DIR=/opt/IBM/db2/V10.5
export DB2DIR
DB2INSTANCE=db2inst1
export DB2INSTANCE
INSTHOME=/export/home/db2inst1
export INSTHOME
PATH=$PATH:$INSTHOME/sql1lib/bin:$INSTHOME/sql1lib/adm:
$INSTHOME/sql1lib/misc
export PATH
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INSTHOME/sql1lib/lib
export LD_LIBRARY_PATH
THREADS_FLAG=native
export THREADS_FLAG
```

3. Save your changes.
4. Stop and restart the IBM InfoSphere Information Server engine.

## Configuring the odbc.ini file

The `odbc.ini` file provides information about connecting to databases and database applications that you use. You must configure this file so that InfoSphere DataStage can access ODBC data sources.

### About this task

The `/opt/IBM/InformationServer/Server/branded_odbc/IBM_Tools` directory contains example `odbc.ini` files for each supported platform. Use the contents of these files to create the `odbc.ini` file for the `$DSHOME` directory. The `$DSHOME` variable identifies the InfoSphere DataStage installation directory.

### Procedure

1. Navigate to the `<installation-directory>/Server/DSEngine` directory, where `<installation-directory>` is the directory where you installed InfoSphere Information Server.
2. Run the following command to create a soft link for the sample `.odbc.ini` file to point to the `odbc.ini` file.

```
ln -s odbc.ini.odbc.ini
```
3. Modify the `.odbc.ini` file to match your database environment.

## Configuring the uvodbc.config file

You specify the ODBC data source name (DSN) for each database in the `uvodbc.config` file.

### About this task

A copy of the `uvodbc.config` file exists in the `$DSHOME` directory and each project directory. For example, the default path for projects on Linux is `/opt/IBM/InformationServer/Server/Projects`.

By default, IBM InfoSphere DataStage searches the current project directory for a `uvodbc.config` file and, if it finds one, uses this file instead of `$DSHOME`. This file arrangement allows you to configure data sources that are known to some projects but not others.

### Procedure

Use this syntax to add entries to the `uvodbc.config` file:

```
<DSN_name>  
DBMSTYPE = ODBC
```

`DSN_name` is the ODBC DSN. This name must be the same name that is specified in the `.odbc.ini` file. The space on each side of the equal sign is required.

In the `uvodbc.config` file, comment lines begin with a number sign (`#`). Comments that are not in the comment block at the top of the file cause processing errors.

Add comments only to the comment block that is at the top of the `uvodbc.config` file.

The following example shows a typical `uvodbc.config` file:

```
[ODBC DATA SOURCES]  
<localuv>  
DBMSTYPE = UNIVERSE  
network = TCP/IP  
service = uvserver  
host = 127.0.0.1  
<Sybase1>
```

```
DBMSTYPE = ODBC
<Sybase2>
DBMSTYPE = ODBC
<Oracle8>
DBMSTYPE = ODBC
<Informix>
DBMSTYPE = ODBC
<DB2>
DBMSTYPE = ODBC
```

## Creating and binding a DB2 package to a DSN

To use a DB2 package, you must bind it to each DB2 data source name (DSN).

### Procedure

1. Open a terminal window, and change to the `$DSHOME/./branded_odbc/bin` directory. The `$DSHOME` variable identifies the InfoSphere DataStage installation directory. The default directory is `/opt/IBM/InformationServer/Server/DSEngine`.
2. Enter the following command:  

```
bind22 dsn
```

**dsn** is the ODBC data source name.
3. Specify your user ID and password, if you are prompted to provide them.

## Configuring other ODBC drivers

You can use ODBC drivers from other vendors to connect to data sources. However, you can use only one ODBC Manager, so you cannot use drivers from other vendors if you use the InfoSphere DataStage drivers.

### Procedure

1. Install the ODBC manager and drivers. Use the documentation that the driver vendor provides.
2. Configure the computer to connect to the driver.
  - a. In the `dsenv` file, add the required environment variables.
  - b. In the `odbc.ini` file, set up the connection to the database or to the database client application. You must specify the Data Source Name (DSN), driver name, and login information.
  - c. In the `uvodbc.config` file, specify the DSN to use to connect to the database.
3. Include the shared library `libodbc.sl` for the ODBC Manager in the InfoSphere DataStage load library path. (On HP-UX, the suffix is `so`.) Edit the `dsenv` file and ensure that the path to the ODBC Manager precedes the path to `$DSHOME/./branded_odbc/lib` in the shared library path environment variable.
4. After you install and configure the driver, test the connectivity.

## Results

To use the DataDirect drivers again, remove the path to the ODBC Manager from the `dsenv` file, and ensure that the InfoSphere DataStage ODBC driver library precedes any other ODBC driver libraries.

## Testing ODBC connectivity

After you configure ODBC connectivity that you defined for InfoSphere DataStage, test the configuration to verify your connection to the data source.

## Procedure

1. Change directories to the \$DSHOME directory and set up the environment by running the following command:  
`. ./dsenv`
2. Enter the following command to start the IBM InfoSphere Information Server engine shell:  
`./bin/dssh`
3. Enter the following command to log in to the project:  
`LOGTO project_name`  
  
where *project\_name* is case-sensitive.
4. Enter the following command to obtain a list of available data source names:  
`DS_CONNECT`
5. Enter the following command to test the connection:  
`DS_CONNECT dsn`  
  
where *dsn* is the data source name of the connection to test.
6. Enter your user name and password to connect to the specified database.
7. Enter `.Q` to quit the connection.

---

## Chapter 3. Module 1: Opening and running the sample job

The tutorial includes a sample job that you explore to better understand basic concepts about jobs. You open the sample job, explore the stages that comprise the job, and then compile and run the job.

The sample job consists of two stages: a Sequential File stage and a Data Set stage. The Sequential File stage extracts data from a comma-separated values file and writes the data to the Data Set stage. The data that is written to the Data Set stage is used by later modules in the tutorial.

### Learning objectives

After you complete the lessons in this module, you understand how to complete the following tasks:

- Start the InfoSphere DataStage and QualityStage Designer (the Designer client) and attach a project
- Open an existing job
- Compile a job so that it is ready to run
- Open the InfoSphere DataStage and QualityStage Director (the Director client) and run a job
- View the results of the job

**Note:** For the remainder of the tutorial, the short name is used when referring to the InfoSphere DataStage components that you interact with. For example, the InfoSphere DataStage and QualityStage Designer is referred to as the Designer client.

### Time required

This module takes approximately 30 minutes to complete.

### Prerequisites

Ensure that you have InfoSphere DataStage and QualityStage User permissions.

---

## Lesson 1.1: Opening the sample job

In this lesson, you view a completed job to understand its structure and familiarize yourself with the Designer client.

To get started with using the Designer client, you open the `samplejob` job that is supplied with this tutorial.

### Learn more about the Designer client:

The Designer client is your workbench and your toolbox for building jobs. You use the included tools to create jobs that extract, transform, load, and check the quality of data. The palette contains the tools that form the basic building blocks of your jobs. To design jobs, you add the following objects from the palette to your canvas:

- *Stages* connect to data sources to read or write files and to process data

- *Links* connect the stages that your data flows through
- *Annotations* provide information about the jobs that you create

The Designer client uses a repository where you store the objects that you create as part of the design process. The sample job is an object in the repository that is included with this tutorial. The sample job uses a table definition, which is also an object in the repository. You store other metadata in the repository such as job parameters, parameter sets, and shared containers. Other job designers can reuse these objects to promote reuse across your organization.

## Overview

The sample job reads data from a flat file and writes the output to a data set. Parallel jobs use data sets to store data while the data is being processed. Data sets can be transient and invisible to you, the designer, or you can choose to create persistent data sets. The sample job writes data to a persistent data set, which provides an internal staging area where the data is held until it is written to its ultimate destination in a later module. When designing jobs, you do not have to create a staging area for your data, but the tutorial uses a data set to persist data from one module to the next.

The data that you use in this job is the customer billing information from GlobalCo. You use this data as the input for your transformation job later in this tutorial. The data contains inconsistencies and formatting issues that you fix by applying data typing and transformation functions. The cleansed data becomes the GlobalCo\_billing dimension for the GlobalCo Worldwide star schema. Another developer creates this star schema to combine the GlobalCo billing information and WorldCo billing information.

## Procedure

1. Click **Start > Programs > IBM InfoSphere Information Server > IBM InfoSphere DataStage and QualityStage Designer**.
2. In the Attach window, type your user name and password.
3. Select the Tutorial project from the **Project** list, then click **OK**.
4. In the New window that opens when you start the Designer client, click **Cancel** to close it, because you are opening an existing job.
5. In the repository tree, expand the Tutorial folder, then double-click the samplejob job. All of the objects that you use for the tutorial are in this folder. You create additional folders later in the tutorial to store your jobs, parameter sets, and table definitions.

The job opens in the Designer client canvas. The following graphic shows the samplejob job.



Figure 1. The samplejob job is composed of two stages that are connected by a single link.

In this lesson, you opened your first job. Now that the job is open, you can begin exploring the stages that the job comprises.

---

## Lesson 1.2: Viewing and compiling the sample job

In this lesson, you explore each of the stages in the `samplejob` job to understand how the job is designed. You then compile the job to prepare it to run on your computer.

### Overview

The `samplejob` job uses a Sequential File stage to read data from a flat file and a Data Set stage to write data to the staging area. The two stages are joined by a link, which carries data between the stages. The metadata that flows between the two stages on the link was defined when the job was designed. When you run the job, the metadata moves along this link.

### Tasks in this lesson

1. “Exploring the Sequential File stage”
2. “Exploring the Data Set stage” on page 14
3. “Compiling the sample job” on page 14

## Exploring the Sequential File stage

Open the Sequential File stage to understand the stage properties. The `GlobalCo_billTo_flat` Sequential File stage is the source. It specifies the source file that for the `samplejob` job, and defines the columns that are sent to the `GlobalCoBillTo_ds` Data Set stage.

1. In the sample job, double-click the `GlobalCo_billTo_flat` Sequential File stage.  
The stage editor opens to the Properties tab of the Output page. All parallel job stages have a Properties tab that you use to specify the actions that occur when you run the job.
2. Look at the File property under the **Source** category.  
You use this property to specify the file that the stage reads when the job runs. In the sample job, the File property points to a file called `GlobalCo_BillTo.csv`. You specify the directory that contains this file when you run the job.  
The name of the directory is specified by a job parameter named `#tutorial_directory#`. You use job parameters to specify variable information at run time, such as a file name or directory name, which simplifies development and processing. Job parameters are specified by number signs (`#`).
3. Click the First Line is Column Names property under the **Options** category.  
In the sample job, this property is set to True because the first line of the `GlobalCo_BillTo.csv` file contains the names of the columns in the file. The remaining properties use default values.
4. Click the **Format** tab.  
The Format tab looks similar to the Properties tab, but the properties that you set describe the format of the flat file that the stage reads. The `GlobalCo_billTo_flat` stage uses a comma-separated values (CSV) file (Delimiter=comma), which means that each field within a row is separated by a comma. The Record delimiter string = DOS format value indicates that the file can be read on AIX, HP-UX, and Solaris systems.
5. Click the **Columns** tab.

You define column metadata for the stage in the Columns tab. The column metadata defines the data that flows along the link to the GlobalCoBillTo\_ds Data Set stage. The GlobalCo\_BillTo.csv file contains many columns, all of which have the data type VarChar. As you work through the tutorial, you apply stricter data typing to these columns to cleanse the data.

6. Click **View Data** in the upper right corner of the stage editor window.
  - a. In the Resolve Job Parameter window, in the **Value** field, specify the name of the directory in which the tutorial data was imported, and click **OK**. You must specify the directory path when you view data in a stage or run a job.
  - b. In the Data Browser window, click **OK**.

A window opens that shows the first 100 rows of the GlobalCo\_BillTo.csv file. Each row represents a customer record that contains a customer number, name, address, contact information, and important sales data.

You write these records to the target file that is specified in the GlobalCoBillTo\_ds Data Set stage.
  - c. Click **Close** to close the Data Browser window.
7. Click **OK** to close the Sequential File stage editor.

## Exploring the Data Set stage

Open the Data Set stage to understand the stage properties. The GlobalCoBillTo\_ds Data Set stage is the target, and specifies the file where the data set is written to.

1. Double-click the GlobalCoBillTo\_ds Data Set stage.

The stage editor opens to the Properties tab of the Input page. Data Set stages are used to store data that is used by another job. Data sets, denoted by a .ds extension, are the internal format for transferring data inside parallel jobs.
2. Click the File property under the **Target** category.

This property points to the GlobalCo\_BillTo.ds file, which is the output file that the stage writes data to at run time. When the job runs, you specify the directory where you want to save this file. Like the previous stage, this stage uses a parameter for the directory where the tutorial files exist.
3. Click the **Columns** tab.

The column metadata for this stage is the same as the column metadata for the Sequential File stage because you are writing the data without transforming it. The Columns tab indicates that the data that is written to the GlobalCo\_BillTo.ds file when you run the job.

The Data Set stage editor does not have a Format tab because the data set does not require any formatting. If you click the **View Data** button, you see a message that no data exists. The data is created after you compile and run the job.
4. Click **OK** to close the Data Set stage editor.

## Compiling the sample job

Compile the samplejob job so that you can run it.

1. From the Designer client menu bar, click **File > Compile**.

As the job compiles, the Compile Job window updates with messages from the compiler.
2. When the Compile Job window displays a message that the job is compiled, click **OK**.

In this lesson, you explored a basic data extraction job that reads data from a flat file and writes it to a data set. Now that the job is compiled, you run the job and view the output.

---

## Lesson 1.3: Running the sample job

In the previous lesson, you compiled the sample job. Now that the sample job is compiled, you use the Director client to run the job and view the job log.

### Overview

You run the job from the Director client. The Director client is the operating console that you use to run and troubleshoot jobs that you develop in the Designer client. You also use the Director client to run fully developed jobs in the production environment.

After you run the job, you view the job log to debug any errors that might exist.

### Tasks in this lesson

1. “Running the job”
2. “Viewing the output”

## Running the job

Thus far, you have been working in the Designer client. You use the Director client to run the samplejob job.

1. From the Designer client menu bar, click **Tools > Run Director**.

Because you are logged in to the Tutorial project through the Designer client, you can start the Director client without reentering your InfoSphere DataStage credentials.

In the Director client, the sample job has a status of **Compiled**, which means that the job is ready to run.

2. Select the samplejob job in the right pane of the Director client, and click **Job > Run Now** from the menu bar.
  - a. In the Job Run Options window, specify the path of your project. For example, C:\IBM\InformationServer\Server\Projects\Tutorial.
  - b. Click **Run**. The job status changes to **Running**.
3. When the job status changes to **Finished**, click **View > Log**.
4. Examine the job log to view the information that the Director client reports as it runs a job.

The job log shows the timestamp of when each event occurred, the type of event, and in which stage or operator the event occurred. All of the messages are either Control or Info messages because the job ran successfully. If errors occurred, Warning messages might appear in the job log that explain the errors. If a job fails, an Aborted message is written to the job log to explain the failure.

5. Click **Project > Exit** to close the Director client.

## Viewing the output

After you run the samplejob job, you view the output to ensure that the target data is in the format that you expect.

1. In the Designer client, open the samplejob job.
2. Open the GlobalCoBillTo\_ds stage.

3. In the stage editor, click **View Data**.
4. In the Data Browser window, click **OK** to accept the default settings. A window opens that shows up to 100 rows of the data written to the data set.  
The data in the data set was not altered when you ran the job. You wrote the data to a data set so that you can use this data as input later in the tutorial.  
The data contains inconsistencies and errors that you fix in your transformation job.
5. Click **Close** to close the Data Browser window.
6. Click **OK** to close the GlobalCoBillTo\_ds stage.

## What's next

You ran the samplejob job and viewed the output data. Now that you understand how to run a data extraction job, you can start creating your own job. The next module guides you through the process of creating another extraction job. You create two lookup file sets for the GlobalCo country code information and special handling codes. You use of these lookup tables in a later module to enhance the GlobalCo customer records.

In this module, you completed the following tasks:

- Started the Designer client
- Opened the samplejob job and located objects in the repository tree
- Explored stages and viewed the data that each stage represents
- Started the Director client from the Designer client
- Ran the samplejob job and viewed the job log
- Examined the output data was written to the GlobalCo\_BillTo\_ds data set

---

## Chapter 4. Module 2: Designing your first job

You learned how to open, compile, and run the sample job. However, this job was prebuilt, so it is time to learn how to design your own job.

The job that you design in this tutorial reads two flat files and populates two lookup tables. To better understand their customer demographics, GlobalCo Worldwide wants to know where each customer lives and what language they speak. Therefore, you create a lookup table that includes a code that identifies which country the customer is from, in addition to a code that indicates the language of each customer.

GlobalCo Worldwide also wants to know the interval at which customers receive shipments. Understanding the frequency and volume of shipments helps GlobalCo Worldwide to streamline their delivery operations and adjust production based on peak delivery times. To include this data, you create a second lookup table that includes a special handling code that is used in Module 3. This code corresponds to a text description that describes the interval of when each customer receives a delivery.

### Learning objectives

After completing the lessons in this module, you understand how to complete the following tasks:

- Add stages and links to a job
- Specify properties for stages and links to determine how they function when you run the job
- Import metadata for columns
- Add job parameters and create parameter sets
- Enhance your knowledge of compiling and running jobs

### Time required

This module takes approximately 90 minutes to complete.

---

## Lesson 2.1: Creating your job

The first step in designing a job is to create a job and save it to a directory in your project.

### Overview

You create a parallel job and save it to a new folder in the Tutorial folder in the repository tree. A job consists of stages that are linked together, which describe the flow of data from a data source to a data target. A stage is a graphical representation of the data itself, or of a transformation that is performed on that data.

After you create your job, you add stages and link them together.

### Tasks in this lesson

1. "Creating a new job"
2. "Adding stages and linking them"

## Creating a new job

In the Designer client, create a new job and save it to a directory in your project.

1. Log in to the Designer client.
2. From the menu bar, click **File > New**.
3. In the New window, select the Jobs folder in the left pane and then click **Parallel job** in the right pane.
4. Click **OK**. An empty job canvas opens.
5. Click **File > Save**.
  - a. In the Save Parallel Job As window, right-click on the Tutorial folder and click **New > Folder** from the shortcut menu.
  - b. Name the new folder GCWW tutorial, where GCWW represents GlobalCo Worldwide.
  - c. Right-click the folder that you created and click **New > Folder**.
  - d. Name the new folder Jobs. You save all of the jobs that you create in this folder to keep it separate from the work of other developers.

In the Repository, ensure that your folder structure matches the following hierarchy:

```
[ - ] Tutorial
    [ - ] GCWW tutorial
        [ + ] Jobs
```

- e. In the Item name field, type Dev\_CustData\_Codes\_Lookup. This name indicates that this job is a development job that includes a lookup for customer data codes.

**Tip:** Always use meaningful names for your jobs, stages, and links. Using a consistent naming convention with meaningful names ensures that your jobs are easy to understand and locate within a directory.

- f. Ensure that the **Folder path** field contains the path \Tutorial\Jobs, and then click **Save**.

## Adding stages and linking them

Add stages to your job and link them together. The Lookup File Set stage that you create is used as input later in this tutorial.

Ensure that the Dev\_CustData\_Codes\_Lookup job that you created is open in the Designer client

1. Add a Lookup File Set stage to your job.
  - a. In the Palette, expand the **File** category.
  - b. Click **Lookup File Set**, then click a blank area on the right side of your job canvas.
  - c. Right-click the stage and select **Rename**.
  - d. Rename the stage as CountryCode\_LKP\_FS and press the Enter key. As stated earlier, using meaningful names is important. This name indicates that this stage is a lookup file set (LKP\_FS) that includes country code data.
2. Add a Sequential File stage to your job.
  - a. In the Palette, expand the **File** category.

- b. Click **Sequential File**, then click a blank area to the left of the CountryCode\_LKP\_FS stage.
  - c. Rename the Sequential File stage as CountryCode\_In\_SF. This name indicates that this stage is a sequential file that contains country code input data.
3. Link the stages that you added to your job.
  - a. In the Palette, expand the **General** category.
  - b. Click **Link**.
  - c. Place your cursor over the CountryCode\_LKP\_FS stage, left-click, and drag your cursor to the CountryCode\_In\_SF stage.
  - d. Rename the link as lnk\_CountryCode\_In.
4. From the toolbar, click **File** > **Save** to save your job.

Your job should look similar to the one in the following figure:



Figure 2. Your first job is composed of a Sequential File stage that writes to a Lookup File Set stage.

In this lesson, you created your first job, added stages and links to your job, and saved your job in the repository.

---

## Lesson 2.2: Specifying metadata

In this lesson, you specify the column metadata for the job that you created in Lesson 2.1. As you develop additional parts of your job, recompile and run your job. This practice ensures that each part of your job is functioning correctly before you design the next part of your job.

### Overview

The first part of your job reads a comma-separated values (CSV) file that contains a series of customer numbers, a corresponding code that identifies the country in which the customers are located, and another code that specifies the customer's language. Your job reads the CSV file and writes the contents to a lookup table in a Lookup File Set stage. This lookup table is used by a subsequent job to populate a dimension table.

### Tasks in this lesson

1. "Specifying column metadata"
2. "Specifying lookup properties and running the job" on page 21

## Specifying column metadata

To specify the actions that occur when you run the job, you edit the properties for your stages. You begin by describing the column metadata that flows from the CountryCode\_In\_SF stage.

1. Double-click the CountryCode\_In\_SF stage. The stage editor opens to the **Properties** tab of the Output page.

- a. Under the **Source** category, click the **File** property.
  - b. In the **File** field, click the browse icon () , and then click **Browse for file**.
  - c. In the file selection window, navigate to the directory where the files for the tutorials exist.
  - d. Select the CustomerCountry.csv file, then press the Enter key to submit the value for the **File** property.
  - e. Under the **Options** category, click the **First Line is Column Names** property.
  - f. Change the value for this property to **True**. This value indicates that the row that contains column names is dropped when the job reads the source file.
2. Click the **Format** tab.
    - a. Click **Record level**.
    - b. In the **Available properties to add** section, click **Record delimiter string**.
    - c. From the **Record delimiter string** drop-down menu, click **DOS format**. This setting ensures that the source file can be read on an AIX, HP-UX, Solaris, or Linux computer.
  3. Click the **Columns** tab.
    - a. Enter the following column metadata. Leave blank any columns whose values are not indicated in the following table.

**Important:** Column names are case-sensitive.

Column name	Key	SQL type	Length	Nullable	Description
CUSTOMER_NUMBER	↙	Char	7	No	Customer identification number
COUNTRY		Char	2	No	Code that identifies the country that the customer lives in
LANGUAGE		Char	2	No	Code that identifies the language of the customer

Because the CustomerCountry.csv file contains only three columns, you enter the column definitions manually. If a file contains many columns, it is less time consuming and more accurate to import the column definitions directly from the data source. You learn how to import table definitions later in this tutorial.

- b. Click **Save** to save the column definitions that you specified as a table definition object in the repository. The definitions can then be reused in other jobs.
- c. In the Save Table Definition window, enter the following information:

*Table 2. Table definition details*

Parameter	Value
Data source type	CSV
Data source name	CustomerCountry.csv
Table/file name	lnk_CountryCode_In

Table 2. Table definition details (continued)

Parameter	Value
Long description	Table definition for country codes source file

- d. Click **OK** to specify the location where you want to save the table definition.
- e. In the **Save Table Definition As** window, right-click the Tutorial file and select **New > Folder**.
- f. Rename the folder Table definitions. You store all of your table definitions in this folder.
- g. Name your table definition `lnk_CountryCode_In`, then click **Save**.
4. In the stage editor window, click **View Data**, then click **OK** in the Data Browser window to use the default settings.  
The data from the `CustomerCountry.csv` file is displayed in the Data Browser window. The data is mapped to the column metadata that you specified.
5. Click **Close** to close the Data Browser window.
6. Click **OK** to close the stage editor window.
7. Save your job.

A table definition icon () is added to the `lnk_CountryCode_In` link. This icon specifies that the link now has column metadata assigned to it.

## Specifying lookup properties and running the job

You defined the metadata that is sent from the `CountryCode_In_SF` stage to the `CountryCode_LKP_FS` stage. Now, you configure the lookup properties for this metadata.

1. Open the `CountryCode_LKP_FS` stage.
2. Click **Lookup Keys**, then select the **Key** property from the **Available Properties to add** list.
3. In the **Key** drop-down menu, select **CUSTOMER\_NUMBER**, then press the Enter key.

You specified that the `CUSTOMER_NUMBER` column is the lookup key for the lookup table that you are creating.

4. Click the **Lookup File Set** property under the **Target** category.
  - a. In the **Lookup File Set** field, click the browse icon () , then click **Browse for file**. The file browser window opens.
  - b. In the file browser window, navigate to the directory where you want to save the file set, such as `C:\IBM\InformationServer\Server\Projects\Tutorial\Tutorial_files`.
  - c. In the **File name** field, enter a name for the file set that you are creating. For example, `CountryCode_lookup.fs`.

**Important:** Ensure that you include the `.fs` extension, which indicates that this file is a file set.

- d. Press the Enter key to submit the path name for the lookup file set.
5. In the `CountryCode_LKP_FS` stage editor, click **OK**.
6. Save your job and then click the compile button () to compile your job.

7. After the job compiles successfully, open the Director client and run your job.
8. After the job runs successfully, click the job log icon ().  
In the job log, you see an informational message that looks similar to the following message. This message indicates the number of rows that were processed successfully, in addition to the number of failed records.  
CountryCode\_In\_SF,0: Import complete;  
1998 records imported successfully, 0 rejected.

You designed and ran your first job. Now, you import metadata and assign it to the links that connect the stages in your job.

---

## Lesson 2.3: Importing metadata

In Lesson 2.2, you manually entered the column metadata that you wanted to import. In this lesson, you import column metadata directly from the data source that your job reads. You store the metadata in the InfoSphere DataStage repository where it can be used in other jobs.

### Overview

Importing metadata from source files reduces the incidence of error and simplifies the mapping of metadata. In this lesson, you add stages to the job that you designed in Lesson 2.2. The stages that you add are similar to the ones that you added previously. The stages read a comma-separated values (CSV) file that contains code numbers and corresponding special delivery instructions. The contents are written to a lookup table that is used by a job later in this tutorial. The finished job contains two separate data flows, each writing data to a separate Lookup File Set stage.

### Tasks in this lesson

1. "Importing metadata into your repository"
2. "Loading column metadata from the repository" on page 23

## Importing metadata into your repository

You import column definitions from the CSV file that contains special delivery instructions. You then save the column definitions as a table definition in the repository. You can import metadata without opening a job in the Designer client, but importing metadata when your job is already open is a good practice to ensure that you correctly map the metadata.

1. In the Designer client, click **Import > Table Definitions > Sequential File Definitions**.
2. In the Import Meta Data window, import the SpecialHandling.csv file.
  - a. In the **Directory** field, click the browse icon () to navigate to the directory where the source files for the tutorial exist. For example, C:\IBM\InformationServer\Server\Projects\Tutorial\Tutorial\_files.
  - b. In the **File Type** drop-down menu, select **Comma Separated (\*.csv)**.
  - c. In the **Files** list, select the SpecialHandling.csv file.
  - d. In the **To folder** field, click the browse icon (). In the Select Folder window, select the Tutorial\Table definitions folder that you created previously, and then click **OK**.

- e. In the Import Meta Data window, click **Import** to import metadata from the SpecialHandling.csv file.
3. In the Define Sequential Metadata window, select the **First line is column names** check box.  
This selection indicates that the row that contains column names is dropped when the job reads the source file.
4. Click the **Define** tab. Select the SPECIAL\_HANDLING code as the lookup key.
5. Click **OK** to close the Define Sequential Metadata window.
6. Click **Close** to close the Import Meta Data window.

## Loading column metadata from the repository

You specify the column metadata that a stage uses by loading the metadata from a table definition in the repository.

Open your Dev\_CustData\_Codes\_Lookup job.

In this part of the lesson, you consolidate the job design skills that you learned and load the column metadata from the table definition that you imported.

1. Add a Sequential File stage to your job and rename the stage as SpecHandlingCode\_In\_SF. Position this stage directly beneath the CountryCode\_In\_SF stage.
2. Add a Lookup File Set stage to the right of the SpecHandlingCode\_In\_SF and rename the stage as SpecHandlingCode\_LKP\_FS.
3. Link the two stages that you added to your job. Rename the link as lnk\_SpecHandling\_In. Your job design should look similar to the one in the following figure:



Figure 3. Your updated job contains two Sequential File stages that write to two Lookup File Set stages

4. Configure the SpecHandling\_In\_SF stage.
  - a. Specify the SpecialHandling.csv file as the source file for the stage.
  - b. Click the **Format** tab.
  - c. Select the **Record level** property, then select the **Record delimiter string** property from the **Available properties to add list**.

- d. Specify **DOS format** from the Record delimiter string drop-down menu. This setting ensures that the file can be read if the engine tier is installed on an AIX, HP-UX, Linux, or Solaris computer.
- e. Click the **Columns** tab.
- f. In the Columns tab, click **Load**.
- g. In the Table Definitions window, select the SpecialHandling.csv table definition that you created earlier, then click **OK**.
- h. In the Select Columns window, ensure that all of the columns are listed in the "Selected columns" pane, then click **OK**.
- i. Click **OK** to close the SpecHandling\_In\_SF stage editor.

A table definition icon () is added to the lnk\_SpecHandling\_In link, indicating that you added table definitions to this link.

5. Open the SpecHandlingCode\_LKP\_FS stage.
  - a. Specify a lookup file set for the target file. Name the file SpecHandlingCode.fs
  - b. Select the Lookup keys property, then select the **Key** property from the "Available properties to add" list.
  - c. Specify the SPECIAL\_HANDLING\_CODE
  - d. Click **OK** to close the SpecHandlingCode\_LKP\_FS stage editor.
6. Save, compile, and run your Dev\_CustData\_Codes\_Lookup job.

You expanded your job, created another lookup file set, and imported table definitions to use in your job. Now, you create parameters to specify the location of source files for each stage in your job. You reuse this parameters throughout the remainder of this tutorial.

---

## Lesson 2.4: Adding parameters

In the previous lesson, you specified the location of source files for each stage in your job. You specified that two comma separated values files are used as input, and that two lookup files are used as the target files. In this lesson, you replace the path names for these files with job parameters. You use these parameters to pass values for variables into your job at run time.

### Overview

Specifying the value of the parameter each time that you run the job ensures that you use the correct resources, such as the source files and target files. When you use parameters in your jobs, you promote reuse and create a better job design.

Similar to other objects in your job, you save each parameter in the project repository.

### Tasks in this lesson

1. "Defining job parameters"
2. "Adding job parameters to your job" on page 25
3. "Running your job with parameters" on page 26

## Defining job parameters

Define the parameters that the job uses when it runs. When you run the job, you specify values for each of these parameters.

Ensure that the Dev\_CustData\_Codes\_Lookup job that you created is open in the Designer client.

1. From the menu bar in the Designer client, click **Edit > Job Properties**.
2. In the Job Properties window, click the **Parameters** tab.
  - a. Double-click the first cell in the grid to add a new row.
  - b. Add the following information to create four job parameters:

Parameter Name	Prompt	Type	Help text
CountryCode_In_Source	Path name of the country codes source file	Pathname	Specify the path name of the .csv file that contains the country code definitions
CountryCode_LKP_Target	Path name of the country codes lookup file set	Pathname	Specify the path name of the file set for the country code lookup table
SpecHandlingCode_In_Source	Path name of the special handling codes source file	Pathname	Specify the path name of the .csv file that contains the special handling code definitions
SpecHandlingCode_LKP_Target	Path name of the special handling codes lookup file set	Pathname	Specify the path name of the file set for the special handling lookup table

- c. Click **OK** to close the Job Properties window.
3. Save your job to save the parameters that you created.

## Adding job parameters to your job

Now that you have defined parameters for your job, you incorporate them into your job design.

1. Open the CountryCode\_In\_SF stage.
  - a. Click the **File** property in the Source category.
  - b. Delete the path name that you entered previously.
  - c. In the File field, click the browse icon , then click **Insert job parameter**.
  - d. Select CountryCode\_In\_Source from the list of available parameters, then press the Enter key.
 

The name of the parameter is enclosed by number signs (#). This change indicates that the job requests the name of the file when you run the job.
  - e. Click **OK** to close the CountryCode\_In\_SF stage editor.
2. Open the CountryCode\_LKP\_FS stage.
  - a. Delete the path name that you entered in the Lookup File Set field.
  - b. Specify the CountryCode\_LKP\_Target parameter in the Lookup File Set field.
  - c. Click **OK** to close the CountryCode\_LKP\_Target stage editor.
3. Open the SpecHandlingCode\_In\_SF stage.
  - a. Delete the path name that you entered in the File field.
  - b. Specify the SpecHandlingCode\_In\_Source parameter in the File field.

- c. Click **OK** to close the SpecHandlingCode\_In\_SF stage editor.
4. Open the SpecHandlingCode\_LKP\_FS stage.
  - a. Delete the path name that you entered in the Lookup File Set field.
  - b. Specify the SpecHandlingCode\_LKP\_Target parameter in the Lookup File Set field.
  - c. Click **OK** to close the SpecHandlingCode\_LKP\_FS stage editor.
5. Save and compile your job.

## Running your job with parameters

You run your job with the parameters that you created. When you run the job, you are prompted to supply values for the parameters.

1. Open the Director client from the Designer client.
2. In the Director client, select your job and click the run icon ().
  - a. In the Job Run Options window, click the browse icon () to browse for the source files and target files.
  - b. Click **Run** to run your job.

You defined job parameters to represent the file names in your job and specified values for these parameters when you ran the job. To further simplify your job design, you combine these parameters into a parameter set in the next lesson.

---

## Lesson 2.5: Creating a parameter set

In the previous lesson, you created parameters for your job. To further simplify your job design and runtime operations, you create a parameter set and store it in the project repository. You can then reuse the parameter set in other job designs.

### Overview

When you create the parameter set, you enter a set of default values for the parameters that are input at run time.

You use parameter sets to define job parameters that you might reuse in other jobs. When you want to use these parameters in a job, you insert them into the job properties from the parameter set. You can create parameter sets from existing job parameters, or you can specify the job parameters when you create a new parameter set.

You can also define different sets of values for each parameter set. Parameter sets are stored as files in the InfoSphere DataStage server installation directory and are available to use in your job designs or when you run jobs that use these parameter sets. If you make changes to a parameter set object, the changes are reflected in job designs that use this object when the job is compiled.

**Important:** The parameters that a job is compiled with are available when the job is run. However, if you change the design after the job is compiled, the job links to the updated version of the parameter set.

### Procedure

Ensure that the Dev\_CustData\_Codes\_Lookup job that you created is open in the Designer client.

1. Create a parameter set.
  - a. From the Designer client menu, click **Edit > Job Properties**.
  - b. In the Job Properties window, click the **Parameters** tab.
  - c. Hold the Ctrl key, then click each of the parameters that you created in the previous lesson.
  - d. Click **Create Parameter Set**.
  - e. In the Parameter Set window, in the **Parameter Set name** field, type a name for your parameter set, such as CustData\_Codes\_Lookup\_parmset.
  - f. Type a short description for your parameter set, such as This parameter set includes parameters that relate to source files and target files for a customer data lookup job.

Including short descriptions for your parameter sets helps other developers to quickly understand the purpose of your parameter set and when to use it.
  - g. Click the **Values** tab.
  - h. In the Value File name field, type a name for your parameter set value file, such as CustData\_Codes\_Lookup\_1.
  - i. For each of your job parameters, specify the default path name for the file that the parameter represents. For example, the value for the CountryCode\_In\_Source parameter might be C:\IBM\InformationServer\Server\Projects\Tutorial\Tutorial\_files\CustomerCountry.csv.
  - j. Click **OK**.
  - k. In the Save Parameter Set As window, right-click the Tutorial folder and create a new folder named Parameter sets.
  - l. Click **Save** to save your parameter set in the folder that you created.

A message displays asking if you want to replace the selected parameters with the parameter set that you created. Click **No**.

**How can I modify my parameter set?:** If you modify any parameters in your parameter set, expand the Tutorial\Parameter sets folder that you created and double-click your parameter set. You can add new parameters, modify existing parameters, and change values for any parameters that are included in the parameter set.

If another developer imports your job and wants to use different source files, the developer opens your parameter set, modifies the values for each of the parameters, and then runs the job with the update source data.

2. Click **OK** to close the Job Parameters window.
3. Save and compile your job.

## What's next

You created a parameter set that is available for another job that you create later in this tutorial. The current job continues to use individual parameters rather than the parameter set.

Now that you know how to develop a simple job, you can develop a more complex job. In the next module, you develop a transformation job that adds data typing to the GlobalCo billing data to fix inconsistencies and improve the formatting of the output data.

In this lesson, you completed the following tasks:

- Created your first job and added stages and links to your job

- Set stage properties that determine what actions occur when you run your job
- Imported metadata to use throughout your job
- Created a parameter set from a set of existing job parameters
- Specified a set of default values for the parameters in the parameter set
- Created a parameter set that you use in a subsequent job

---

## Chapter 5. Module 3: Transforming data

You developed a job that writes data from a source file to a target file. Now that you understand the basics of job design, you learn how to design a job that transforms data.

Your transformation job reads from the GlobalCoBillTo\_ds data set that was the target in the sample job in Module 1 of this tutorial. Your transformation job cleanses this data, transforms it by dropping unnecessary columns, and trims extraneous data.

### Learning objectives

After completing the lessons in this module, you understand how to complete the following tasks:

- Use a Transformer stage to transform data
- Capture rejected data
- Combine data by using a Lookup stage
- Complete multiple transformations in a single job

### Time required

This module takes approximately 60 minutes to complete.

---

## Lesson 3.1: Designing your transformation job

You design and run a job that transforms part of the GlobalCoBillTo\_ds data, then writes the results to a Data Set stage.

### Overview

The source data for the sample job, GlobalCo\_BillTo.csv, contains many columns. The dimension table that you create later in this tutorial requires only a subset of these columns, so you use the Transformer stage to remove some of the unnecessary columns. You also add more strict data typing for the remaining columns to impose quality controls on your data. To trim unnecessary characters, you enter a function in the Transformer stage. Using these capabilities and functions to cleanse your data ensures that your output is in the format that you want.

**Learn more about processing stages:** The transformation job that you design uses a Transformer stage, but the Designer client includes other processing stages that can transform data. For example, you can use a Modify stage in your job if you want to change only the data types in a data set. Several of the processing stages can drop data columns as part of their processing. However, you use the Transformer stage because you require a transformation function that you can customize.

### Tasks in this lesson

1. “Creating your transformation job” on page 30
2. “Configuring the Data Set stages” on page 30

3. "Configuring the Transformer stage" on page 31
4. "Running your transformation job" on page 33

## Creating your transformation job

To create a transformation job, you add a Transformer stage between your source and target.

1. Create a parallel job and name it `Dev_CustData_TrimAndStrip`.
  2. Save the job in the `\Tutorial\Jobs` folder that you created in a previous lesson.
  3. Add a Data Set stage to your job.
    - a. In the Designer client Palette, click **File**.
    - b. Click **Data Set**, then click a blank area on your job canvas.
    - c. Rename the Data Set stage as `GlobalCo_BillTo_In_DS`.
  4. Add a Transformer stage to your job.
    - a. In the Designer client Palette, click **Processing**.
    - b. Click **Transformer**, then click a blank area on your job canvas to the right of the `GlobalCo_BillTo_In_DS` stage.
    - c. Rename the stage as `GlobalCo_BillTo_TFM`.
  5. Right-click the `GlobalCo_BillTo_In_DS`, drag your cursor to the `GlobalCo_BillTo_TFM` stage, then release the mouse button to link the stages. This method of linking stages is faster and easier than selecting a Link from the Palette.
  6. Rename the link as `lnk_Full_BillTo_In`.
  7. Add another Data Set stage to the right of the `GlobalCo_BillTo_TFM` stage.
  8. Rename the Data Set stage as `GlobalCo_BillTo_Out_DS`.
  9. Link the `GlobalCo_BillTo_TFM` stage to the `GlobalCo_BillTo_Out_DS` stage.
  10. Rename the link as `lnk_Stripped_BillTo_Out`.
- Your job should look similar to the one in the following graphic:

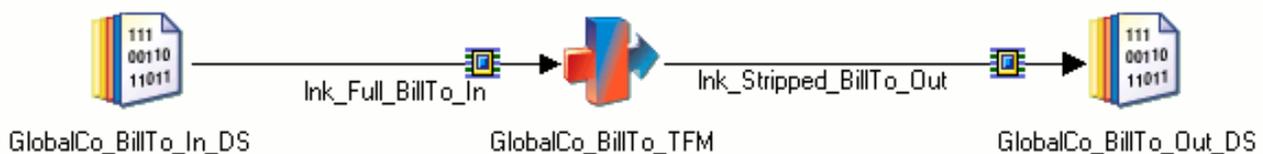


Figure 4. Your job includes a Transformer stage that adds data typing to your data before writing the data to a data set.

## Configuring the Data Set stages

Configure both Data Set stages and then load the column metadata for each stage by using a table definition.

1. Configure the first Data Set stage.
  - a. Open the `GlobalCo_BillTo_In_DS` stage and set the **File** property to point to the `GlobalCoBillTo.ds` data set that sample job wrote to in Module 1.
  - b. Click **OK** to close the stage editor.
2. Add a table definition to your job.
  - a. In the Repository window, expand the `\Tutorial` folder.

- b. Select the GlobalCoBillToSource table definition and drag it to the lnk\_Full\_BillTo\_In link. A table definition icon () is added to the link to show that the table definition is now associated with the link. You imported this table definition when you imported the samplejob job. This method of associating table definitions with your jobs is faster and simpler than loading the table definitions from the stage editor.
3. View the data in the Data Set stage.
  - a. Open the GlobalCo\_BillTo\_In\_DS stage.
  - b. In the stage editor, click **View Data**, then click **OK** to open the Data Browser window.
 

The customer records in the data set are displayed. After you add table definitions to your jobs, view the data to verify that column metadata is correct.
  - c. Click **Close** to close the Data Browser window.
  - d. Click **OK** to close the stage editor.
4. Open the GlobalCo\_BillTo\_Out\_DS stage.
  - a. Set the **File** property to point to the data set that you are creating. For example, if you name your data set GlobalCo\_BillTo\_Out.ds, you include the full path name to that file: C:\IBM\InformationServer\Server\Projects\Tutorial\Tutorial\_files\GlobalCo\_BillTo\_Out.ds.
  - b. Click **OK** to close the stage editor.
5. Save your job.

## Configuring the Transformer stage

Define the transformation operations that occur when you run your job.

1. Open the GlobalCo\_BillTo\_TFM stage.
2. Specify the columns that flow through the Transformer stage at run time.
  - a. In the left pane, select the following columns in the lnk\_Full\_BillTo\_In link.

CUSTOMER\_NUMBER  
 CUST\_NAME  
 ADDR\_1  
 ADDR\_2  
 CITY  
 REGION\_CODE  
 ZIP  
 TEL\_NUM  
 REVIEW\_MONTH  
 SETUP\_DATE  
 STATUS\_CODE

- b. Drag these columns to the lnk\_Stripped\_BillTo\_Out link in the right pane. By selecting a subset of columns, you specify the columns that flow through the Transformer stage at run time. The remaining columns are dropped.

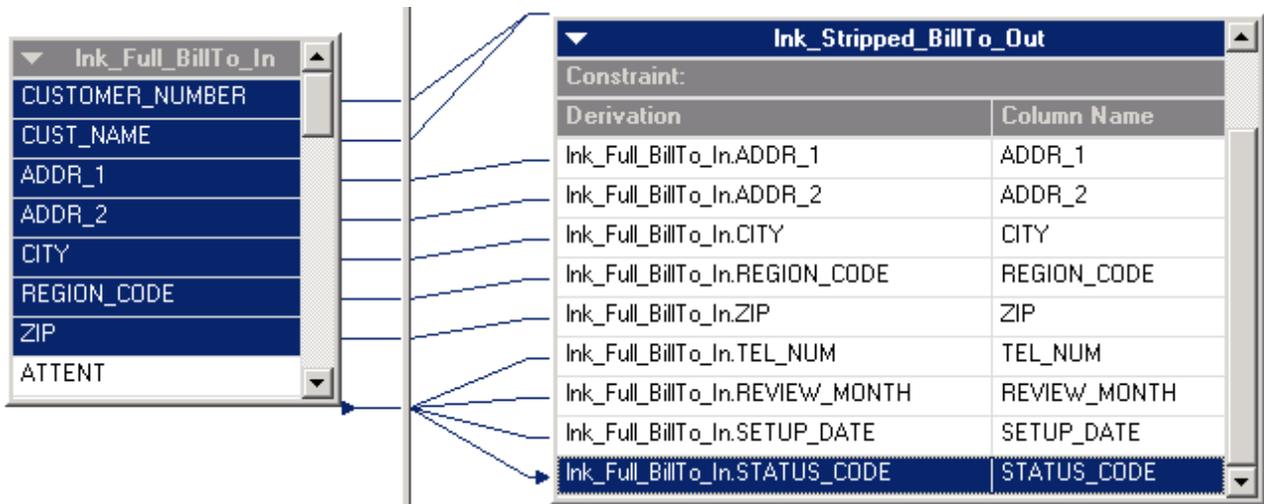


Figure 5. You map the columns from the input link to the output link. This mapping defines the column metadata that flows into and out of the Transformer stage.

3. In the lower right pane, enter the following values for the **SQL type**, **Length**, and **Description** column definitions for the Ink\_Stripped\_BillTo\_Out link.

By specifying stricter data typing, you can better diagnose inconsistencies in your source data when you run the job. Use the default values for any fields that are not specified in the following table.

Column name	SQL type	Length	Description
CUSTOMER_NUMBER	Char	7	Unique customer number
CUST_NAME	VarChar	30	Full name of the customer
ADDR_1	VarChar	30	Street name and number of the customer address
ADDR_2	VarChar	30	Additional street information, such as unit number
CITY	VarChar	30	City that the customer resides in
REGION_CODE	Char	2	Region that the customer resides in, if applicable
ZIP	VarChar	10	Postal code where the customer resides
TEL_NUM	VarChar	10	Telephone number for the customer
REVIEW_MONTH	VarChar	2	Month that the customer record is reviewed internally
SETUP_DATE	VarChar	12	Date when the customer record was created in the data set

Column name	SQL type	Length	Description
STATUS_CODE	Char	1	Code that indicates whether the customer record is active, on hold, or inactive

4. Add a derivation to the CUSTOMER\_NUMBER column.

The expression editor opens.

- a. In the lnk\_Stripped\_BillTo\_Out link, double-click the **Derivation** field for the CUSTOMER\_NUMBER column, and delete the derivation.

- b. Click the browse icon () , then click **Function**.

- c. Expand the **String** category and select **Trim**. The derivation field is populated with a basic trim function.

- d. Click the browse icon () , click **Input Column**, then select the CUSTOMER\_NUMBER column.

- e. Delete the parameters after the column name, and then type the following syntax.

```
' ', 'A'
```

- f. Press the Enter key to submit the derivation. The derivation for the CUSTOMER\_NUMBER column looks like the following derivation:

```
Trim(lnk_Full_BillTo_In.CUSTOMER_NUMBER,' ','A')
```

This derivation specifies that extraneous spaces are deleted from the CUSTOMER\_NUMBER column on the lnk\_Full\_BillTo\_In link. The spaces are removed before the data is written to the CUSTOMER\_NUMBER column on the lnk\_Stripped\_BillTo\_Out link. You can enter the derivation manually, but using the built-in functions helps to reduce errors and provides the structure of each function.

5. Click **OK** to close the stage editor.
6. Open the GlobalCo\_BillTo\_Out\_DS stage, then click the **Columns** tab.  
The data set inherited the columns that you input to the Transformer stage.
7. Save and then compile your job.

## Running your transformation job

Previously, you ran jobs from the Director client. In this lesson, you run your job from the Designer client. This method of running a job is useful when you are developing jobs because you do not leave the Designer client. Click **View > Job Log** to view the job log file.

1. From the Designer client menu, click **Diagram > Show performance statistics**. This view provides additional information as the job runs, such as the number of rows that are transferred, and the number of rows that are processed per second.
2. Click **File > Run**, then click **Run** in the Job Run Options window.  
As your job runs, performance statistics for the link are updated, and the links change color to show the status. If the job runs successfully, the links turn green. The number of rows that are processed are displayed under the link names.

**Note:** If you receive an error that the job failed because of an error in the Transformer stage, your computer might not have a C++ compiler installed. For

more information about installing a compiler, see “Setting up a C++ compiler” in the *IBM InfoSphere Information Server Planning, Installation, and Configuration Guide*.

3. When the job finishes, open the Director client. You can view the job log in the Director client even when you run the job from the Designer client.
  - a. Select the Dev\_CustData\_TrimAndStrip job.

- b. Click the job log icon () to view the job log.

- c. In the job log, double-click the event that reads main\_program: This step has 2 datasets: (...).

The Message field contains the following information, which helps you understand the components that your job is composed of.

**This section highlights the data sets that are used in your job.**

main\_program: This step has 2 datasets:

**The first data set, ds0, is linked to the GlobalCo\_BillTo.ds source file.**

```
ds0: {C:/IBM/InformationServer/Server/Projects/Tutorial/Tutorial_files/GlobalCo_BillTo.ds
     eAny=>eCollectAny
     op0[2p] (parallel APT_CombinedOperatorController:GlobalCo_BillTo_In_DS)}
```

**The second data set, ds1, is linked to the GlobalCo\_BillTo\_Out.ds target file.**

```
ds1: {op0[2p] (parallel APT_CombinedOperatorController:GlobalCo_BillTo_Out_DS)
     =>
     C:/IBM/InformationServer/Server/Projects/Tutorial/Tutorial_files/GlobalCo_BillTo_Out.ds}
```

**Your job uses one operator, which is the GlobalCo\_BillTo\_TFM Transformer stage.**

It has 1 operator:

```
op0[2p] {(parallel APT_CombinedOperatorController:
          (GlobalCo_BillTo_In_DS)
          (APT_TransformOperatorImplV0S1_Dev_CustData_TrimAndStrip_GlobalCo_BillTo_TFM
           in GlobalCo_BillTo_TFM)
          (GlobalCo_BillTo_Out_DS)
         )
         on nodes (
           node1[op0,p0]
           node2[op0,p1]
         )
       }
```

**Your job runs on two nodes: node1 and node 2. |**

It runs 2 processes on 2 nodes.

- d. Click **Close** to close the Event Details window.

You designed and configured your first transformation job. You filtered the column metadata to narrow your data for processing, and then included a function to cleanse your data.

In the next lesson, you add a Lookup stage that references a data set that you created earlier. Including this lookup ensures that only matching records are written to the target data set.

---

## Lesson 3.2: Using a Lookup stage in your transformation job

In the previous lesson, you designed a simple transformation job. In this lesson, you build a more complex job that combines data from two different tables.

### Overview

You base your job on the transformation job that you created in the previous lesson. You then add a Lookup stage that references data from a lookup table that you created in Lesson 2.2. You defined the key columns for the lookup tables when you created the lookup file sets.

You can configure Lookup stages to search for data in a Lookup file set or in a relational database. The Lookup stage has a stream input and a reference input. The stage uses one or more key columns in the stream input to search for data in a

reference table. The data from the reference table is then added to the stream output. When you use lookup file sets, you must specify the lookup key column when you define the file set.

Using a Lookup stage to create a lookup is one method of combining data. You can also combine data in a parallel job by using a Join stage. If your job uses a large reference table, combining data by using a Join stage is faster than using a Lookup stage. The reference table that you use in your job is small, so a Lookup stage is preferred. The Lookup stage is most efficient where the data that is looked up fits into physical memory.

### Tasks in this lesson

1. “Creating a lookup job”
2. “Configuring the Lookup File Set stage” on page 36
3. “Configuring the Lookup stage” on page 37

## Creating a lookup job

Create a new job that includes the stages and links from the Dev\_CustData\_TrimAndStrip job.

1. Create a parallel job and name it Dev\_CustData\_CleansePrepare.
2. Save the job in the \Tutorial\Jobs folder that you created in a previous lesson.
3. Open the Dev\_CustData\_TrimAndStrip job, and then drag your cursor around the stages and links.
  - a. From the menu, click **Edit > Copy**.
  - b. In the Dev\_CustData\_CleansePrepare job, click **Edit > Paste** to paste the stages and links that you copied.

You are finished working with the Dev\_CustData\_TrimAndStrip job, so close that job.
4. In the Dev\_CustData\_CleansePrepare job, delete the GlobalCo\_BillTo\_Out\_DS stage. You will replace this stage with a different stage.
5. Add a Lookup stage to the Dev\_CustData\_CleansePrepare job.
  - a. From the Palette, click **Processing**.
  - b. Click **Lookup**, and then drop the stage on your canvas to the right of the GlobalCo\_BillTo\_TFM stage.
  - c. Rename the stage as GlobalCo\_Country\_LKP.
  - d. Drag the lnk\_Stripped\_BillTo\_Out link to the GlobalCo\_Country\_LKP stage. When you connect the link, the associated column metadata moves with the link. This column metadata can flow between the GlobalCo\_BillTo\_TFM stage and the GlobalCo\_Country\_LKP stage.
6. Add a Lookup File Set stage to your job.
  - a. From the Palette, click **File**.
  - b. Click **Lookup File Set**, and drop the stage on your canvas above the GlobalCo\_Country\_LKP stage.
  - c. Rename the stage as GlobalCo\_CountryCodes\_FS.
  - d. Link the GlobalCo\_CountryCodes\_FS stage to the GlobalCo\_Country\_LKP stage.
  - e. Rename the link lnk\_CountryCode\_Ref. The link is displayed as a dotted line, which indicates that the link is a reference link.
7. Add a Data Set stage to the right of the GlobalCo\_Country\_LKP stage.
  - a. Rename the stage as GlobalCo\_CountryLookup\_DS.

- b. Link the GlobalCo\_Country\_LKP stage to the GlobalCo\_CountryLookup\_DS stage.
- c. Rename the link as lnk\_CountryCode\_Out.

Your job looks like the following job:

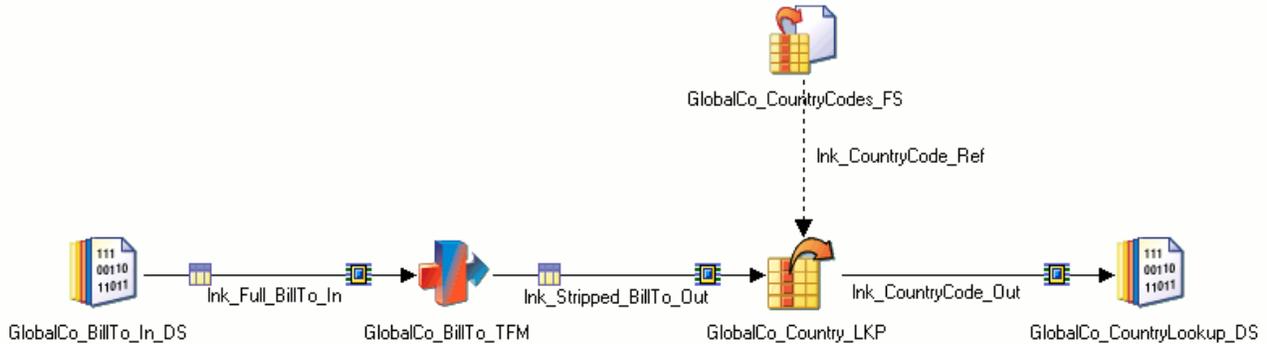


Figure 6. Your job includes a Lookup stage, which references country data against the customer data that is the output from the Transformer stage.

## Configuring the Lookup File Set stage

In the previous section of this lesson, you copied stages from the Dev\_CustData\_TrimAndStrip job to the Dev\_CustData\_CleansePrepare job. These stages are already configured, so you need to configure only the stages that you added. The first stage that you configure is the GlobalCo\_CountryCodes\_FS stage. This stage uses the lookup file set that you created in Lesson 2.2. In addition, you use the parameter set that you created in Lesson 2.5 to populate the parameters that the job uses.

1. Add a parameter set to your job.
  - a. Open the Job Properties window for the Dev\_CustData\_CleansePrepare job, and then click the **Parameters** tab.
  - b. Click **Add Parameter Set**.
  - c. In the Open window, expand the Tutorial\Parameter sets folder.
  - d. Select the CustData\_Codes\_Lookup\_parmset parameter set that you created in Lesson 2.5, then click **OK**.
  - e. Click **OK** to close the Job Properties window.
2. Configure the GlobalCo\_CountryCodes\_FS stage.
  - a. Open the GlobalCo\_CountryCodes\_FS stage. Under the **Source** category, click the **Lookup File Set** property.
  - b. In the Lookup File Set field, click the browse icon (  ), then click **Insert Job Parameter**.
  - c. Select the CountryCode\_LKP\_Target parameter, then press the Enter key. This parameter associates the stage with the SpecialHandlingCode.fs file set.
  - d. Click the **Columns** tab.
  - e. Load the lnk\_CountryCode\_In table definition that you created in a previous lesson.
  - f. Close the GlobalCo\_CountryCodes\_FS stage.

## Configuring the Lookup stage

In the Lookup stage, you specify the data that you want to combine. When you created the `Ink_CountryCode_In` table definition in Lesson 2.2, you specified that the `CUSTOMER_NUMBER` column is the key for the lookup. You use this key to compare the source data to the data in your lookup file set.

1. Map the lookup metadata from the input columns to the output columns.

- a. Open the `GlobalCo_Country_LKP` stage. In the left pane, click the title bar of the `Ink_Stripped_BillTo_Out` link, then drag your cursor to the `Ink_CountryCode_Out` link.

The column metadata is mapped from the input link to the output link.

- b. In the left pane, click the `COUNTRY` column in the `Ink_CountryCode_Ref` link, then drag your cursor to the `Ink_CountryCode_Out` link.

When the value of the incoming `CUSTOMER_NUMBER` column matches the value of the `CUSTOMER_NUMBER` column of the lookup table, the corresponding `COUNTRY` column is added to that row of data. In the stage editor, the columns are mapped from the `Ink_Stripped_BillTo_Out` link to the `Ink_CountryCode_Out` link. The `CUSTOMER_NUMBER` column is used as a key in the Lookup stage to reference customer data.

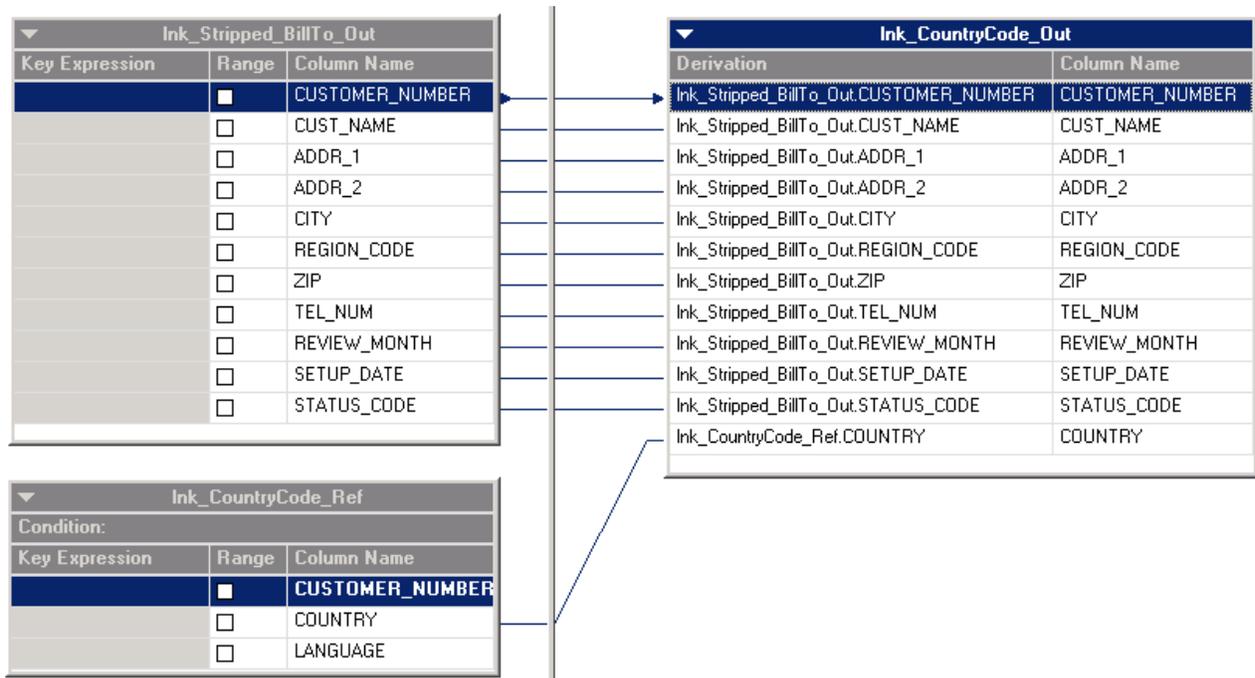


Figure 7. The columns are mapped from the input link to the output link.

- c. In Lookup Stage Conditions window, in the `Ink_CountryCode_Ref` link, double-click **Condition**.

- d. In the Lookup Failure column, select **Continue**.

This option specifies that if a value from the `CUSTOMER_NUMBER` column in the `Ink_Stripped_BillTo_Out` link does not match any value in the `CUSTOMER_NUMBER` column in the `Ink_CountryCode_Ref` link, the job continues processing the `CUSTOMER_NUMBER` column.

- e. Click **OK** to close the Lookup Stage Conditions window.
- f. Close the `GlobalCo_Country_LKP` stage.

2. Open the GlobalCo\_CountryLookup\_DS stage and specify a file name for the data set target. For example, you might name the file GlobalCo\_CustData\_Cleansed.ds.
3. Save, compile and run the job. When you run the job, the Job Run Options window displays all of the parameters in the parameter set.
  - a. In the Job Run Options window, select CustData\_Codes\_Lookup\_1 from the list next to the parameter set name.  
The parameters values populate with the path names that you specified when you created the parameter set.
  - b. Click **Run** to run the job.
4. After the job runs successfully, open the GlobalCo\_CountryLookup\_DS stage, and then click **View Data**.  
The customer data is displayed in the Data Browser window.

In this lesson, you removed extraneous characters from the GlobalCo customer data and included a lookup so that only matching records were written to the target data set. By cleansing the customer data, you prepared the information to be merged with the WorldCo customer data.

---

## Lesson 3.3: Capturing rejected data

In this lesson, you add a reject link to your job to capture rejected data when the job runs.

### Overview

In Lesson 3.2, you specified an option in the Lookup stage to continue processing customer data if the lookup operation fails. Any rows containing values in the CUSTOMER\_NUMBER column that are not matched in the lookup table are bypassed, and the COUNTRY column for that row is set to NULL. GlobalCo Worldwide wants to capture these rejected customer records so that their customer advocate team can analyze and update the records.

To accommodate this request, you specify that any customer numbers that do not match an entry in the lookup table are written to a reject file.

### Procedure

1. Open the Dev\_CustData\_CleansePrepare job.
2. Add a Sequential File stage to your job under the GlobalCo\_Country\_LKP stage.
  - a. Rename the Sequential File as GlobalCo\_CountryCode\_Rej.
  - b. Link the GlobalCo\_Country\_LKP stage to the GlobalCo\_BillTo\_Rejects stage.  
Because the GlobalCo\_Country\_LKP stage already has an output link, the reject link is shown as a dashed line.
  - c. Rename the link as lnk\_CountryCode\_Rej.  
Your job looks like the following job.

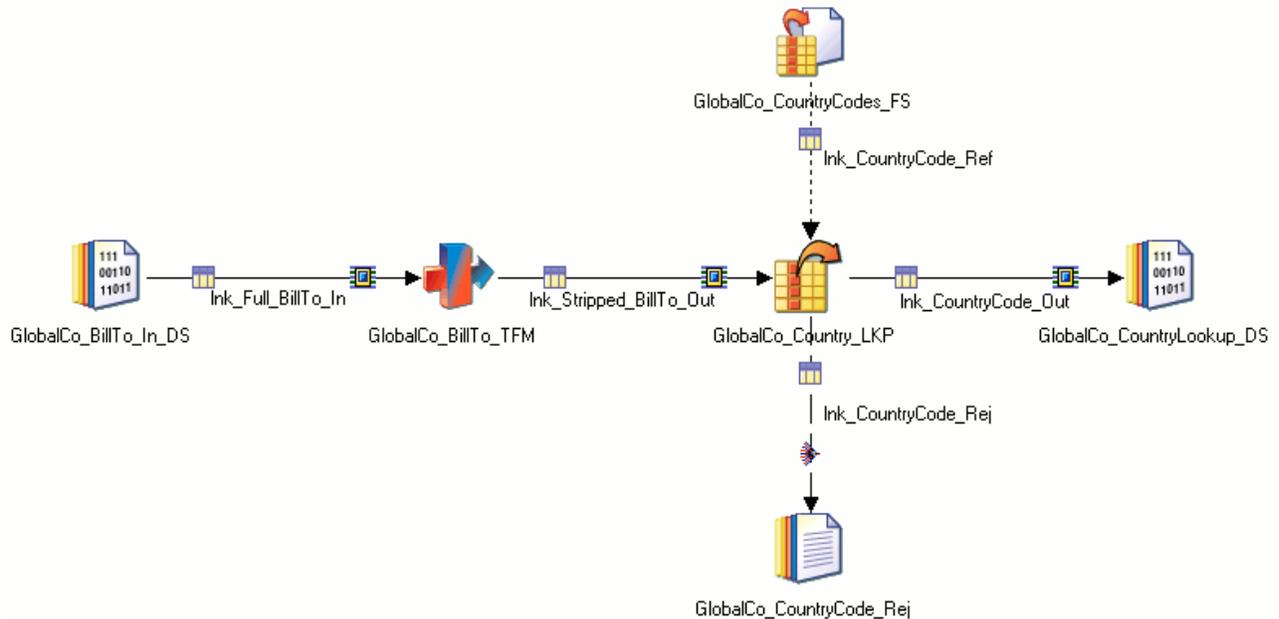


Figure 8. Your job includes a reject link to collect any customer records that do not match an entry in the lookup table.

3. Open the GlobalCo\_Country\_LKP stage.
  - a. In the Ink\_CountryCode\_Ref link, double-click **Condition**.
  - b. In the Lookup Stage Conditions window, select **Reject** in the Lookup Failure column.
 

This option specifies that if a value from the CUSTOMER\_NUMBER column in the Ink\_Stripped\_BillTo\_Out link does not match any value in the CUSTOMER\_NUMBER column in the Ink\_CountryCode\_Ref link, the row is sent to the GlobalCo\_CountryCode\_Rej stage.
  - c. Close the Lookup Stage Conditions window and the GlobalCo\_Country\_LKP stage editor.
4. Open the GlobalCo\_CountryCode\_Rej stage and specify a file name for the text file where the rejected records are written to. For example, C:\IBM\InformationServer\Server\Projects\Tutorial\_files\GlobalCo\_CountryCode\_Rej.txt.
5. Save, compile, and run the Dev\_CustData\_CleansePrepare job.
6. After the job runs successfully, right-click the GlobalCo\_CountryCode\_Rej stage and click **View Ink\_CountryCode\_Rej data**.
 

The Data Browser window opens to show the customer records that did not meet the lookup criteria.
7. Right-click the GlobalCo\_CountryLookup\_DS stage and click **View Ink\_CountryCode\_Out data**.
 

The Data Browser window opens to show the customer records that were written to the data set.

You captured the rejected customer records that did not meet the lookup criteria. These rejected records can be combined with records from WorldCo so that the GlobalCo Worldwide customer advocate team can analyze the combined data. The team can investigate inaccuracies in the data to determine why the customer numbers had no matching country code in the lookup table. Fixing these inconsistencies ensures that the GlobalCo Worldwide billing records are accurate and complete.

Some of the dates in the `SETUP_DATE` column are overloaded with extraneous data that must be removed. In addition, the `STATUS_CODE` column does not contain the associated description of what each code means. These problems must be fixed before the customer billing data is written to the GlobalCo Worldwide database.

In the next lesson, you further cleanse the customer billing data. You use another Transformer stage and Lookup stage to remove the extraneous data from the `SETUP_DATE` column and add the descriptions that relate to the `STATUS_CODE` column.

---

## Lesson 3.4: Completing multiple transformations in a single job

In this lesson, you further transform your data to apply business rules, then complete a lookup that uses an additional reference table.

### Overview

When you completed Lesson 3.3, you viewed the output data to review your results. You realized that one of the columns in the `GlobalCo_BillTo.ds` data set contains extraneous data, and that the extraneous data is now in your output data set. Some of the values in the `SETUP_DATE` column contain a special handling code in addition to the date that the account was set up. These values must be separated into the appropriate columns before the data is combined with the WorldCo customer data.

To separate the values, you add transformation logic to extract the special handling code from the `SETUP_DATE` column into a separate column. You then add a lookup operation to look up the text description that corresponds to the code from the lookup table that you populated in Lesson 2.3. You write this description to the output data, and also add a row count.

### Tasks in this lesson

1. “Adding transformation and lookup operations”
2. “Adding transformation logic to your job” on page 41
3. “Configuring the Lookup operation” on page 44

## Adding transformation and lookup operations

To complete multiple transformations in your job, add another Transformer stage and additional Lookup stages.

1. Open the `Dev_CustData_CleansePrepare` job.
2. Delete the `GlobalCo_CountryLookup_DS` stage. You create a new data set for the output.
3. Add a Transformer stage to the right of the `GlobalCo_Country_LKP` stage.
  - a. Rename the stage as `SpecHandling_Extract_TFM`.
  - b. Drag the `lnk_CountryCode_Out` link to the `SpecHandling_Extract_TFM` stage. Because logic remains with the link where you defined it, the columns that you defined flow from the first lookup to the new Transformer stage.
4. Add a Lookup stage to the right of the `SpecHandling_Extract_TFM` stage.
  - a. Rename the stage as `SpecHandling_Code_LKP`. This stage looks up the special handling code to find the corresponding text description.
  - b. Link the `SpecHandling_Extract_TFM` stage to the `SpecHandling_Code_LKP` stage.
  - c. Rename the link as `lnk_SpecHandlingCode_Out`.

5. Add a Lookup File Set stage above the SpecHandling\_Code\_LKP stage.
  - a. Rename the stage as SpecHandling\_Desc\_FS.
  - b. Link the SpecHandling\_Desc\_FS stage to the SpecHandling\_Code\_LKP stage.
  - c. Rename the link as lnk\_SpecHandlingDesc\_Ref.
6. Add a Data Set stage under the SpecHandling\_Code\_LKP stage.
  - a. Rename the stage as GlobalCo\_CustData\_DS.
  - b. Link the GlobalCo\_CustData\_DS stage to the SpecHandling\_Code\_LKP stage.
  - c. Rename the link as lnk\_CustData\_Out.

Your job looks like the following job:

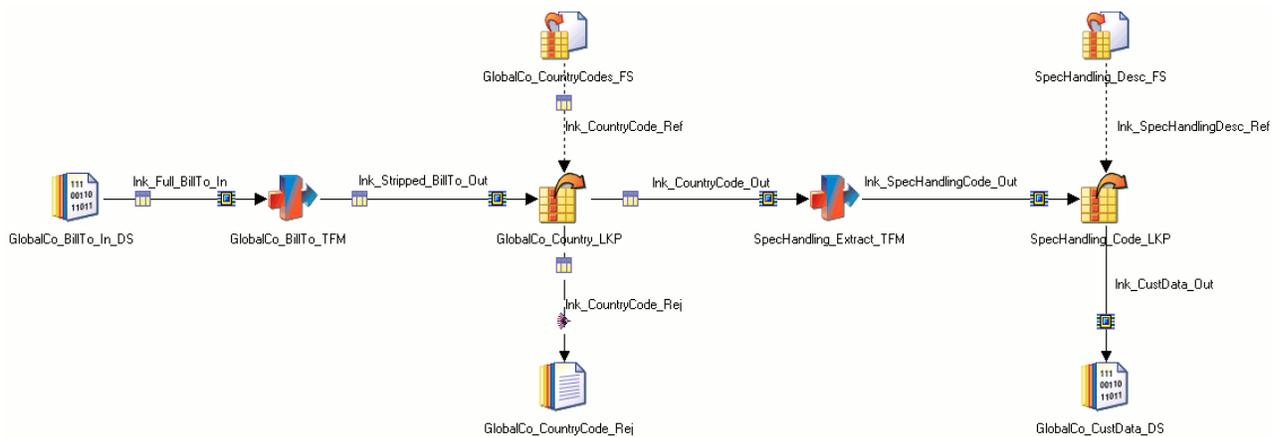


Figure 9. Your job includes a second Transformer stage to further cleanse the GlobalCo customer data.

## Adding transformation logic to your job

Add transformation logic to extract the special handling code, then add a row count to the output data.

1. Open the SpecHandling\_Extract\_TFM stage.
2. In the left pane of the Transformer Stage window, select the following columns in the lnk\_CountryCode\_Out link and drag them to the lnk\_SpecHandlingCode\_Out link in the right pane.

CUSTOMER\_NUMBER  
 CUST\_NAME  
 ADDR\_1  
 ADDR\_2  
 CITY  
 REGION\_CODE  
 ZIP  
 TEL\_NUM

3. In the lower right pane, add the following columns for the lnk\_SpecHandlingCode\_Out link. Use the default values for any fields that are not specified in the following table.

**Important:** The columns that you add are highlighted in red in the lnk\_SpecHandlingCode\_Out link because they do not have valid derivations. You add the derivations to specify how each column is processed by the Transformer stage.

Column name	SQL Type	Length	Nullable	Description
SOURCE	Char	10	No	Specifies the company that the customer data comes from
RECNUM	Char	10	No	Identifies the number of rows for each record
SETUP_DATE	Char	10	Yes	Checks whether the SETUP_DATE column contains a date
SPECIAL_HANDLING_CODE	Integer	10	Yes	Specifies whether the SETUP_DATE column contains a special handling code

4. In the `lnk_SpecHandlingCode_Out` link container, double-click the **Derivation** field for the SOURCE column.
  - a. In the expression editor, type `'GlobalCo':`.
  - b. Move your cursor to the right of the text that you entered, right-click, and click **Input Column**.
  - c. Select the COUNTRY column, then press Enter to submit the derivation.  
This derivation adds the prefix "GlobalCo" to the two-letter country code. For example, when you run your job, a customer record with the country code US becomes GlobalCoUS. The following code shows the derivation for the SOURCE column:  
`'GlobalCo': lnk_CountryCode_Out.COUNTRY`
5. Double-click the **Derivation** field for the REC\_NUM column.
  - a. In the expression editor, type `'GC':`.
  - b. Move your cursor to the right of the text that you entered, right-click, and click **System Variable**.
  - c. Select `@OUTROWNUM`, then press the Enter key to submit the derivation.  
This derivation adds row numbers to your output so that the formatting is easier to read. Also, other developers can refer to customer records by their row number rather than randomly searching through the output. The following code illustrates the derivation for the REC\_NUM column:  
`'GC': @OUTROWNUM`
6. In the Transformer Stage editor toolbar, click the Stage Properties icon ().
  - a. In the Transformer Stage Properties window, click the **Stage Variables** tab.
  - b. Add stage variables. Use the default values for any fields that are not specified in the following table.

Name	SQL type	Precision
SpecHandlingExtract	Char	1
TrimSetupDate	VarChar	10

- c. Click **OK** to close the Transformer Stage Properties window.  
The stage variables that you created are listed in the Stage Variables container.
7. In the Stage Variables container, double-click the **Derivation** field for the SpecHandlingExtract variable.
  - a. Enter the following expression, then press Enter. You can right-click in the expression editor to add columns like you did in step 4.

```
IF LEN(1nk_CountryCode_Out.SETUP_DATE) < 2 THEN 1nk_CountryCode_Out.SETUP_DATE
ELSE FIELD(1nk_CountryCode_Out.SETUP_DATE, ' ',2)
```

This expression specifies that, if the number of characters in the SETUP\_DATE column is less than two, then use the value from the SETUP\_DATE column in the SPEC\_HANDLING\_CODE column. Otherwise, the value of the SETUP\_DATE column is returned, and the two-digit character is extracted from that column and moved to the SPEC\_HANDLING\_CODE column.
  - b. Drag the derivation for the SpecHandlingExtract variable to the Derivation field for the SPEC\_HANDLING\_CODE column in the 1nk\_SpecHandlingCode\_Out link container.  
A line is shown between the stage variable and the column, and the name SpecHandlingExtract is listed in the **Derivation** field. For each row that is processed, the value of the SpecHandlingExtract variable is written to the SPEC\_HANDLING\_CODE column.
8. In the Stage Variables container, double-click the **Derivation** field for the TrimSetupDate variable.
  - a. Type the following expression, then press Enter. You can right-click in the expression editor to add columns like you did in step 4 and step 7.

```
IF LEN(1nk_CountryCode_Out.SETUP_DATE) < 3 THEN '01/01/0001'
ELSE FIELD(1nk_CountryCode_Out.SETUP_DATE, ' ',1)
```

This expression says that, if the SETUP\_DATE column does not contain a date, then set the value of the TrimSetupDate variable to '01/01/0001'. Otherwise, the value of the SETUP\_DATE column is used in the TrimSetupDate variable.
  - b. Drag the derivation for the TrimSetupDate variable to the **Derivation** field for the SETUP\_DATE column in the 1nk\_SpecHandlingCode\_Out link container.  
A line is shown between the stage variable and the column, and the name TrimSetupDate is listed in the **Derivation** field. For each row that is processed, the value of the TrimSetupDate variable is written to the SETUP\_DATE column.
9. Click **OK** to close the Transformer Stage window, and then save your job.

Your stage editor looks like the following stage editor:

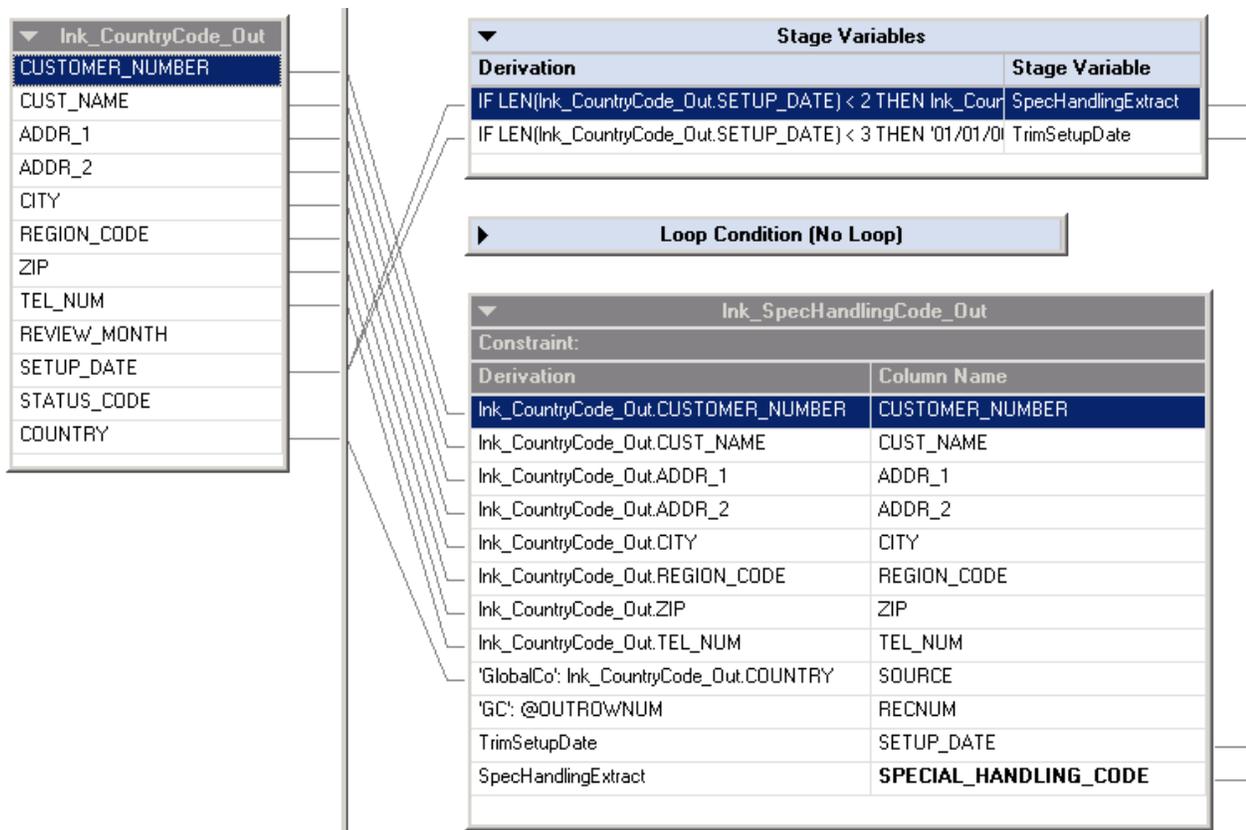


Figure 10. You map the columns from the input link to the output link. The stage variables that you add provide additional transformation logic to fix inconsistencies and errors in the GlobalCo customer data.

## Configuring the Lookup operation

In this exercise, you configure the stages that are required to look up the special handling code and write it to an output data set.

1. Open the SpecHandling\_Desc\_FS stage.
  - a. Set the Lookup File Set property to reference the SpecHandlingCode\_In\_Source parameter.
  - b. Click the **Columns** tab and load the SpecialHandling.csv table parameter.
  - c. Click **OK** to close the stage editor.
2. Open the SpecHandling\_Code\_LKP stage.
  - a. Select all of the columns from the Ink\_SpecHandlingCode\_Out link container and drag them to the Ink\_CustData\_Out link container.
  - b. Select the DESCRIPTION column from the Ink\_SpecHandlingDesc\_Ref link container and drag it to the Ink\_CustData\_Out link container.
  - c. In the lower right pane, type a description for the DESCRIPTION column. For example, Description for the special handling code.
  - d. In the Ink\_SpecHandlingDesc\_Ref link container, double-click the **Condition** bar.
  - e. Select **Continue** in the Lookup Failure field, then click **OK**.

When you inspected the output data at the end of Lesson 3.3, you noticed that few rows in the sample data contain a special handling code. Therefore, you want the job to continue processing data if the rows that do not contain a code are rejected.

Your stage editor looks like the following stage editor:

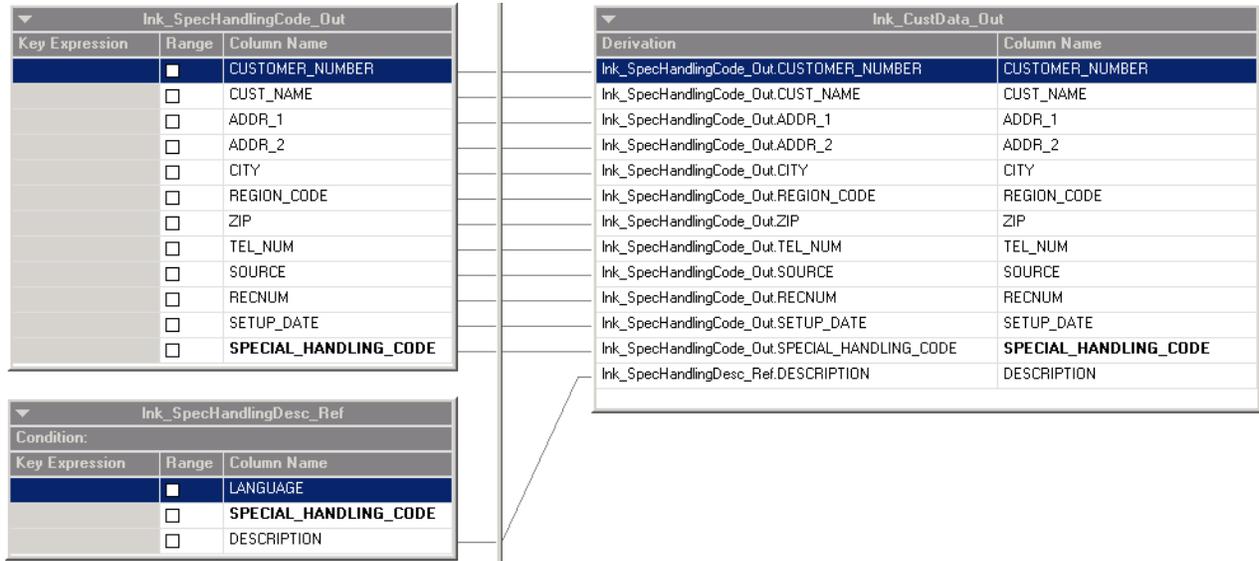


Figure 11. You map the columns from the input link to the output link. The `SPECIAL_HANDLING_CODE` column is used as a key in the Lookup stage to reference customer data.

- f. Click **OK** to close the SpecHandling\_Code\_LKP stage editor.
3. Open the GlobalCo\_CustData\_DS stage.
  - a. In the File field, click the browse icon, then click **Insert job parameter > New**.
  - b. In the Job Properties window, enter the following values for the job parameter.

Parameter name	Prompt	Type	Default Value	Help Text
CustData_Output_dataset	Target file for the customer data output	Pathname	<i>tutorial_folder</i> \GlobalCo_Target_Output.ds	Target file that contains the cleansed GlobalCo customer data

*tutorial\_folder* is the folder where you want to save the files for the tutorial, such as `C:\IBM\InformationServer\Server\Projects\Tutorial\Tutorial_files`.

- c. Click **OK** to close the Job Properties window.
- d. In the File property, specify the `CustData_Output_dataset` parameter, then press the Enter key.
- e. Click **OK** to close the GlobalCo\_CustData\_DS stage editor.
4. Save, compile, and run the SpecHandling\_Extract\_TFM job.
5. After the job runs successfully, right-click the GlobalCo\_CustData\_DS stage and click **View Ink\_CustData\_Out data**.
6. In the Resolve Job Parameters window, click **OK**.

The Data Browser window opens, displaying the cleansed customer data.

In the SOURCE column, the two-digit country code is prefaced by GlobalCo. Including this information is important so that developers can distinguish between the GlobalCo customer data and the WorldCo customer data when the cleansed data is merged.

The SPECIAL\_HANDLING\_CODE column contains a special handling code for each customer record. The corresponding description for each code is listed in the DESCRIPTION column. The shipping department at GlobalCo Worldwide can use this information to determine when each customer receives their delivery.

## What's next

In this lesson, you applied stricter data typing to further transform and cleanse the GlobalCo billing data. To this point, you wrote your output data to data sets. In the following module, you write the output data to a relational database so that other developers in the GlobalCo Worldwide organization can easily access the data.

You completed the following tasks:

- Defined stage variables in a Transformer stage
- Filtered column metadata in a Transformer stage
- Used system variables to generate output column values
- Configured Lookup stages to enhance your output data
- Viewed the job log to better understand the components that your job is composed of

---

## Chapter 6. Module 4: Loading data to a relational database

In this module, you write the GlobalCo\_billing data to a relational database. This database is the final target for the GlobalCo billing data that you transformed and cleansed.

In the previous modules that you completed, you used CSV files and staging files in internal formats, such as data sets and lookup file sets. In this module, you use a relational database as your target.

For these lessons, you use the database that is hosting the tutorial repository. The tutorial supplies scripts that your database administrator runs to create the tables that you need for these lessons.

Because different types of relational databases can be used to host the tutorial repository, the lessons in this module use an ODBC connection. This type of connection ensures that you can follow the instructions regardless of which database type you use.

### Learning objectives

After completing the lessons in this module, you understand how to complete the following tasks:

- Define a data connection object that you use to connect to a database
- Import column metadata from a database
- Write data to a relational database target

### Time required

This module takes approximately 60 minutes to complete.

### Prerequisites

Ensure that your database administrator completed the tasks in “Creating a DSN for the parallel job tutorial table” on page 5. You use the DSN that your database administrator creates to establish an ODBC connection.

---

## Lesson 4.1: Creating a data connection object

In this lesson, you create a data connection object to store the information that is required to connect to the database where you write the cleansed customer billing data.

### Overview

You use this data connection object with an ODBC connector stage to define a connection to the GlobalCo Worldwide data source. You can also use data connection objects to provide the details that are required to connect to a data source and import metadata.

**Learn more about data connection objects:** If you change the details of a data connection while you are designing a job, the changes are reflected in the job

design. However, after you compile your job, the data connection details are fixed in the executable version of the job. Subsequent changes to the job design link to the data connection object and record any changes that you make to that object.

You can create data connection objects directly in the Designer client Repository. Also, you can save the connection details to create data connection objects. Use this option when you are using a connector stage to import metadata. In this lesson, you create a data connection object directly in the Designer client Repository.

## Creating a data connection object

1. In the Designer client Repository, right-click the Tutorial folder, then click **New > Other > Data Connection** from the shortcut menu.
  - a. In the Data Connection window, enter a name for the data connection object. For example, GlobalCo\_billing\_connect
  - b. Enter a short description, such as Connection object for the GlobalCo billing data.
  - c. Enter a long description that describes the details of this connection, such as the name of the database that you are connecting to and the metadata that this connection represents.
2. Click the **Parameters** tab.
  - a. In the **Connect using Stage Type** field, click the browse icon ().
  - b. In the Open window, expand the **Stage Types > Parallel > Database** folder.
  - c. Click **ODBC Connector**, then click **Open**.
  - d. Enter values for each of the Parameters as shown in the following table.

Parameter name	Value
<b>ConnectionString</b>	<i>Name of the DSN that you are using to connect to the database</i>
<b>Username</b>	<i>User name for the DSN that you are using to connect to the database</i>
<b>Password</b>	<i>Password for the Username that corresponds to the DSN</i>

- e. Click **OK** to submit the data connection details.
3. In the Save Data Connection As window, select the Tutorial folder and click **Save**.

You created a data connection object and stored the object in the Repository. In the next lesson, you import column metadata from a database table. You assign the column metadata to the GlobalCo billing data from Lesson 3.4.

---

## Lesson 4.2: Importing column metadata from a database table

In this lesson, you import column metadata from a database table and store it as a table definition object in the repository.

### Overview

In Lesson 2.3, you imported column metadata from a CSV file. In this lesson, you import column metadata from a database table by using an ODBC connector. You use the imported table definition in the next lesson to write the cleansed GlobalCo customer billing data to the GlobalCo Worldwide database.

When you use a connector to import data, the column definitions are saved as a table definition in the Designer client Repository. The table definition is then available to be used by other projects and by other components in the information integration suite.

## Procedure

1. From the Designer client menu, click **Import > Table Definitions > Start Connector Import Wizard**.
2. In the **Connector metadata import** window, select the database connector type that you want to import from and click **Next**.
  - a. In the Connection pane, click **Load**.
  - b. In the **Open** window, expand the Tutorial folder, select the data connection object that you created in Lesson 4.1, then click **Open**.

The Data source, Username, and Password fields populate with the corresponding data that you stored in the data connection object.
  - c. Click **Test connection** to ensure that you can connect to the database. If the connection fails, type the user name and password, and then test the connection again.
  - d. Click **Next** to proceed.
3. On the **Data source location** page, ensure that the host name where the database resides and the database name are specified correctly, then click **Next**.
4. On the **Filter** page, select the schema for the database from the list, then click **Next**. If you are unsure which database schema to use, contact your database administrator.
5. In the Selection page, select the table for the tutorial from the list, then click **Next**. For example, the table that your administrator created might be named TUTORIAL.
  - a. Click **View data**.
  - b. In the Browse data window, ensure that you are importing the GlobalCo billing data.
  - c. Click **Close** to close the Browse data window.
  - d. On the Selection page, click **Next**.
6. In the Confirmation pane, click **Import**.
7. In the Select Folder window, expand the Tutorial folder, select the Table definitions folder, then click **OK**.

The table definition is imported and is listed in the Tutorial\Table definitions folder in the Repository. The table definition has a different icon from the table definitions that you used previously. This icon identifies that the table definition was imported by using a connector, and is available to other projects and suite components.

You imported a table definition from a database by using a connector. The table definition is based on the data connection object that you created in Lesson 4.1.

In the next lesson, you create a job that writes the table definition that you imported to a database. This database is where the GlobalCo billing data and the WorldCo billing data is stored before it is merged.

## Lesson 4.3: Writing data to a database

In this lesson, you use an ODBC connector to write the GlobalCo billing data that you created in Module 3 to an existing table in the database.

### Overview

As part of configuring the tutorial, your InfoSphere DataStage and QualityStage Administrator ran a script to create a target table in a database. In this lesson, you import the column definitions from this table, which forms the `GlobalCo_billing` dimension table in the finished data warehouse. Another developer merges this dimension table with the `WorldCo_billing` dimension table to create the billing information for GlobalCo Worldwide.

**Learn more about connectors:** Connectors are stages that you use to connect to data sources and data targets. Several types of connectors exist, depending on the type of database that you are reading from or writing to. In the Designer client Palette, expand the Database category to view the connectors that you can use. Some connectors have different versions. For example, IBM DB2 databases include a standard database connector and an enterprise connector.

Use connector stages in your jobs to achieve the following advantages over other stages:

- Create job parameters from the connector stage without defining the job parameters in the job properties
- Save connection information that you specify in the stage as a data connection object
- Reconcile data types between sources and targets to avoid runtime errors
- Generate detailed error information if a connector encounters problems when the job runs

### Creating the job

1. Create a new parallel job and name it `GlobalCo_CustData_DB_Write`.
2. Save the job in the `Tutorial\Jobs` folder in the Repository.
3. Add a Data Set stage to your job and name the stage `GlobalCo_BillTo_Source`.
4. Add a connector to your job.
  - a. Expand the Database category in the Palette.
  - b. Click **ODBC Connector**, and add the connector to the right of the `GlobalCo_BillTo_Source` stage.
  - c. Rename the connector as `GlobalCo_BillTo_Target`.
5. Link the stages, and rename the link as `lnk_GlobalCo_BillTo_Out`.

Your job looks like the following job.

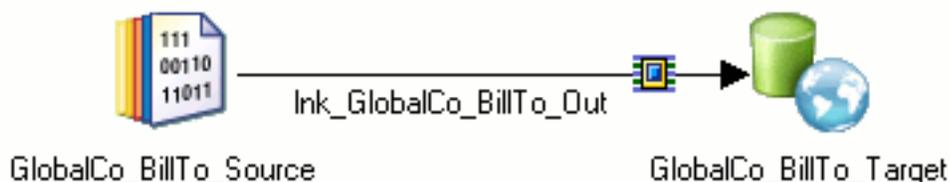


Figure 12. Your job writes the cleansed GlobalCo customer billing data to the shared database.

## Configuring the target

In this exercise, you use the GlobalCo billing data set that you created in Lesson 3.4 as the source for your job.

In this exercise, you use the table definition that you imported in Lesson 4.2. The column definitions are the same as the table definition that you created by editing the Transformer stage and Lookup stage in the job in Lesson 3.4.

1. Open the GlobalCo\_BillTo\_Source stage.
2. Click the **File** property, then create a new parameter that uses the following values. You use the GlobalCo\_Target\_Output.ds data set that you created in Lesson 3.4 as your source data.

Parameter name	Prompt	Type	Default Value	Help Text
CustData_Input_dataset	Source file for the customer data input	Pathname	tutorial_folder\ GlobalCo_Target_ Output.ds	Source file that contains the cleansed GlobalCo customer data

3. Load the column metadata.
  - a. Click the **Columns** tab, then click **Load**.
  - b. In the Table Definitions window, expand the Tutorial\Table definitions folder. Select the database table that you created in Lesson 4.2, then click **OK**.
  - c. In the Select Columns window, ensure that all of the columns are selected, then click **OK**.  
The Columns tab populates with the column metadata that you imported.
4. Click **OK** to close the GlobalCo\_BillTo\_Source stage editor, then save your job.

## Configuring the ODBC connector

In this exercise, you configure the database connector to supply information that is needed to write to the database table. You use the SQL builder to define the SQL statement that is used to write the data to the database when you run the job.

1. Open the GlobalCo\_BillTo\_Target stage.
2. Load the connection properties.
  - a. In the stage editor, click the stage icon.
  - b. In the Connection pane of the Properties tab, click **Load**.
  - c. In the Open window, expand the Tutorial folder and select the data connection object that you created in Lesson 4.1, then click **Open**. For example, your object might be named GlobalCo\_billing\_connect.  
The **Database**, **User name**, and **Password** properties populate with the values from the data connection object.
  - d. In the Connection section of the Properties tab, click **Test** to test the connection.

A message displays saying that the connection was successful. If the connection fails, check the connection details, then test the connection again.

3. Insert and configure the SQL statement.
  - a. Click the **Input** tab.
  - b. Expand the Usage section of the Properties tab and set the **Write mode** to **Insert**. This selection enables the Insert statement field in the SQL category.
  - c. Expand the **SQL** category, click the **Insert statement** row, then click **Build > Build new SQL**.
  - d. In the SQL Builder window, expand the Tutorial\Table definitions folder.
  - e. Expand the Tutorial project, and then select the TUTORIAL table definition that you imported in Lesson 4.2.
  - f. Drag the TUTORIAL table definition to the blank pane to the right of the folder repository.
  - g. In the TUTORIAL table definition, click **Select All**, then drag the column definitions to the **INSERT Columns** pane.  
Your **INSERT Columns** pane includes the columns that you want to write to the target database.

INSERT Columns		
	Insert Column	Insert Value
1	CUSTOMER_NUMBER	ORCHESTRATE.CUSTOMER_NUMBER
2	CUST_NAME	ORCHESTRATE.CUST_NAME
3	ADDR_1	ORCHESTRATE.ADDR_1
4	ADDR_2	ORCHESTRATE.ADDR_2
5	CITY	ORCHESTRATE.CITY
6	REGION_CODE	ORCHESTRATE.REGION_CODE
7	ZIP	ORCHESTRATE.ZIP
8	TEL_NUM	ORCHESTRATE.TEL_NUM
9	SOURCE	ORCHESTRATE.SOURCE
10	RECNUM	ORCHESTRATE.RECNUM
11	SETUP_DATE	ORCHESTRATE.SETUP_DATE
12	SPECIAL_HANDLING_CODE	ORCHESTRATE.SPECIAL_HANDLING_CODE
*		

**INSERT** SQL

Figure 13. You specified the columns that will be written to the target database.

- h. Click the **SQL** tab to view the constructed SQL statement. This statement shows the syntax that inserts the column definitions that you imported.
  - i. Click **OK** to close the SQL Builder window.  
The SQL statement is inserted in the **Insert statement** field.
4. Click **OK** to close the GlobalCo\_BillTo\_Target stage, and then save, compile, and run the job.

You wrote the cleansed GlobalCo customer billing data to the tutorial database table.

## What's next

In this lesson, you used a connector stage to connect to the GlobalCo Worldwide database. You wrote the GlobalCo customer billing data to this database so that other developers can access the data and combine it with the WorldCo customer billing data.

You completed the following tasks:

- Configured a connector stage
- Used a data connection object to supply database connection details
- Used the SQL Builder to define the SQL statement that writes to the database
- Wrote the cleansed GlobalCo customer billing data to the database

You completed the parallel job tutorial for InfoSphere DataStage.

You can now complete any of the following tasks:

- Complete the modules in the *InfoSphere DataStage Advanced Tutorial*.
- Learn more about InfoSphere DataStage by reading the *IBM InfoSphere DataStage and QualityStage Designer Client Guide*.



---

## 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](http://www.ibm.com/able/product_accessibility/index.html).

### **Accessible documentation**

Accessible documentation for InfoSphere Information Server products is provided in an information center. The information center presents the documentation in XHTML 1.0 format, which is viewable in most web browsers. Because the information 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 the information 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. 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 3. 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 <a href="http://www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server">www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server</a>
Software services	You can find information about software, IT, and business consulting services, on the solutions site at <a href="http://www.ibm.com/businesssolutions/">www.ibm.com/businesssolutions/</a>
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 <a href="http://www.ibm.com/account/">www.ibm.com/account/</a>
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 <a href="http://www.ibm.com/training">http://www.ibm.com/training</a>
IBM representatives	You can contact an IBM representative to learn about solutions at <a href="http://www.ibm.com/connect/ibm/us/en/">www.ibm.com/connect/ibm/us/en/</a>



---

## Appendix C. 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 D. 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](http://www.ibm.com/able/product_accessibility/index.html).

### Accessible documentation

Accessible documentation for InfoSphere Information Server products is provided in an information center. The information center presents the documentation in XHTML 1.0 format, which is viewable in most web browsers. Because the information 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 the information 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 E. 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 4. 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 <a href="http://www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server">www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server</a>
Software services	You can find information about software, IT, and business consulting services, on the solutions site at <a href="http://www.ibm.com/businesssolutions/">www.ibm.com/businesssolutions/</a>
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 <a href="http://www.ibm.com/account/">www.ibm.com/account/</a>
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 <a href="http://www.ibm.com/training">http://www.ibm.com/training</a>
IBM representatives	You can contact an IBM representative to learn about solutions at <a href="http://www.ibm.com/connect/ibm/us/en/">www.ibm.com/connect/ibm/us/en/</a>



---

## Appendix F. 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 G. 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/](http://www.ibm.com/software/awdtools/rcf/).
- Send your comments by e-mail to [comments@us.ibm.com](mailto: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 5. 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	<ul style="list-style-type: none"> <li>• Session</li> <li>• Persistent</li> </ul>	User name	<ul style="list-style-type: none"> <li>• Session management</li> <li>• Authentication</li> </ul>	Cannot be disabled
Any (part of InfoSphere Information Server installation)	InfoSphere Metadata Asset Manager	<ul style="list-style-type: none"> <li>• Session</li> <li>• Persistent</li> </ul>	No personally identifiable information	<ul style="list-style-type: none"> <li>• Session management</li> <li>• Authentication</li> <li>• Enhanced user usability</li> <li>• Single sign-on configuration</li> </ul>	Cannot be disabled

Table 5. 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	<ul style="list-style-type: none"> <li>• Session</li> <li>• Persistent</li> </ul>	<ul style="list-style-type: none"> <li>• User name</li> <li>• Digital signature</li> <li>• Session ID</li> </ul>	<ul style="list-style-type: none"> <li>• Session management</li> <li>• Authentication</li> <li>• Single sign-on configuration</li> </ul>	Cannot be disabled
InfoSphere DataStage	XML stage	Session	Internal identifiers	<ul style="list-style-type: none"> <li>• Session management</li> <li>• Authentication</li> </ul>	Cannot be disabled
InfoSphere DataStage	IBM InfoSphere DataStage and QualityStage Operations Console	Session	No personally identifiable information	<ul style="list-style-type: none"> <li>• Session management</li> <li>• Authentication</li> </ul>	Cannot be disabled
InfoSphere Data Click	InfoSphere Information Server web console	<ul style="list-style-type: none"> <li>• Session</li> <li>• Persistent</li> </ul>	User name	<ul style="list-style-type: none"> <li>• Session management</li> <li>• Authentication</li> </ul>	Cannot be disabled
InfoSphere Data Quality Console		Session	No personally identifiable information	<ul style="list-style-type: none"> <li>• Session management</li> <li>• Authentication</li> <li>• Single sign-on configuration</li> </ul>	Cannot be disabled
InfoSphere QualityStage Standardization Rules Designer	InfoSphere Information Server web console	<ul style="list-style-type: none"> <li>• Session</li> <li>• Persistent</li> </ul>	User name	<ul style="list-style-type: none"> <li>• Session management</li> <li>• Authentication</li> </ul>	Cannot be disabled
InfoSphere Information Governance Catalog		<ul style="list-style-type: none"> <li>• Session</li> <li>• Persistent</li> </ul>	<ul style="list-style-type: none"> <li>• User name</li> <li>• Internal identifiers</li> <li>• State of the tree</li> </ul>	<ul style="list-style-type: none"> <li>• Session management</li> <li>• Authentication</li> <li>• Single sign-on configuration</li> </ul>	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](http://www.ibm.com)<sup>®</sup> 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](http://www.ibm.com/legal/copytrade.shtml).

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

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

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<sup>™</sup> 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<sup>Link</sup>, ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS and United States Postal Service. IBM Corporation is a non-exclusive DPV and LACS<sup>Link</sup> licensee of the United States Postal Service.

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



---

# Index

## A

Administrator client 3

## C

column metadata  
importing 22, 48  
combining data  
Lookup stage 34  
customer support  
contacting 57, 63

## D

data  
combining 34  
creating a lookup 34  
rejected data 38  
data connection objects  
creating 47  
Data Set stage  
exploring 13  
database  
creating a target table 5  
database connectors  
configuring 50  
Designer client  
opening a job 11  
starting 11  
designing jobs  
tutorial 17  
Director client  
starting 15  
dsenv file  
configuring 6  
DSN  
configuring ODBC drivers 9  
configuring the dsenv file 6  
configuring the odbc.ini file 8  
configuring the uvodbc.config file 8  
creating 5  
creating a DB2 package 9  
creating for the database 5, 6  
testing ODBC connectivity 10

## J

job design  
adding stages 19  
job log  
viewing data 15  
job parameters  
adding to your jobs 24  
parameter sets 26  
jobs  
compiling 13  
creating 17  
opening 11  
running 15

## L

legal notices 69  
Lookup File Set stage  
referencing data 34  
Lookup stage  
combining data 34

## M

metadata  
importing 22

## O

ODBC  
configuring other drivers 9  
configuring the dsenv file 6  
configuring the odbc.ini file 8  
configuring the uvodbc.config file 8  
creating a DB2 package 9  
testing connectivity 10  
ODBC connection  
creating a DSN 5  
ODBC connections  
creating a DSN  
AIX, HP-UX, Solaris, Linux 6  
Windows 5  
odbc.ini file  
configuring 8

## P

parallel job tutorial  
copying source files 4  
creating a job 17  
designing a job 17  
downloading files 3  
installing 3  
sample job 11  
setting up 3  
parallel jobs  
tutorial 1  
parameter sets  
creating 26  
parallel job tutorial  
importing components 4  
product accessibility  
accessibility 55, 61  
product documentation  
accessing 59, 65  
project  
creating 3

## R

referencing data  
Lookup File Set stage 34  
reject data  
capturing 38

repository objects  
importing table definitions 48

## S

sample job  
overview 11  
Sequential File stage  
exploring 13  
software services  
contacting 57, 63  
source files  
copying to your computer 4  
stages  
adding to your job 19  
exploring stages 13  
Lookup File Set 19  
stage properties 19  
support  
customer 57, 63

## T

table definitions  
importing 22  
target database table  
creating 5  
trademarks  
list of 69  
Transformer stage  
transforming data 29  
transforming data  
Transformer stage 29  
tutorial components  
importing 4  
tutorial files  
downloading 3  
tutorials  
creating parallel jobs 1

## U

uvodbc.config file  
configuring 8

## W

writing to a database  
using database connectors 50







Printed in USA

SC19-4284-00

