

# Data transformation guide for ZipSync

---

Using EPIC ZipSync and Pentaho Data Integration to transform and synchronize your data with xMatters

April 7, 2014

## Table of Contents

Overview	4
About Pentaho	4
Required components	4
System requirements	5
xMatters EPIC client overview	5
ZipSync mode file archive	5
Pentaho Data Integration overview	6
Generic ZipSync/Pentaho sample files overview	6
Process files and directories	7
Sync Data folder	7
Configuration folder	7
Log folder	8
Replacement-files folder	8
Results folder	8
Standard-files folder	8
Generic ZipSync configuration	9
About this example	9
Download and configure the EPIC client	9
Download and install Pentaho	10
Configure the Generic ZipSync support files	12
Run the job	13
Customize the transformation files	14
Configure mirror mode or non-mirror mode	14
External ownership	15
Use your own data instead of sample data	15
Configure EPIC to commit changes	15
ETL flow	15
1. Clean and validate input data	16
2. Build the sites file	17
3. Build the users file	18

4. Build the email devices file	19
5. Build the voice and text devices files	20
6. Build the custom values file	23
7. Gather the statistics from this job run and compare it to the last run	24
Generic ZipSync customization	24
Understanding the job status emails	24

## Overview

This document provides an example how to migrate data from your system into xMatters on demand using the Pentaho Data Integration platform and the xMatters EPIC (Export Plus Import Controller) utility. It is designed to be used in conjunction with a set of sample files contained in `Generic_ZipSync_Pentaho_Process_v7.zip`. This archive contains sample data, Pentaho configuration files, data transformation support files, and EPIC template files. You can use this sample to get started with the data import process and then customize it for your own system.

The sample Pentaho configuration files convert the sample data into a ZipSync archive file that can be used with the EPIC tool. It then runs EPIC in validation mode to test your authentication settings and validate the contents of the ZipSync archive. It does not import the sample data your xMatters system.

After running the initial example, you can to customize the Pentaho configuration files to use data from your system and commit the changes to xMatters.

### About Pentaho

Pentaho is a third-party set of business intelligence, analysis, and data transformation tools. Pentaho provides an open-source community edition of their products and an enterprise edition that contains advanced functionality and support. This document describes how to use the community edition of Pentaho's Data Integration tool. If you require Pentaho support that is not available with the community edition, you may want to consider upgrading to an enterprise subscription.

For more information about Pentaho, or for Pentaho product support, please see <http://pentaho.com>. Note that while xMatters supports its clients with the ETL process generally, it does **not** support Pentaho products.

## Required components

The following components are required to run the examples in this document:

- **xMatters EPIC client:** The EPIC client is an xMatters utility that imports data into xMatters on demand. Use the version of EPIC client that matches your hosted instance. For example, if your xMatters hosted instance is using build 5.5.54, use version 5.5.54 of the EPIC tool.
- **Pentaho Data Integration 5.0.1:** Pentaho is a third-party data integration platform that performs the Extract, Transform, and Load (ETL) process to transform data from your system into the required input format for the EPIC tool.
- **Generic ZipSync/Pentaho sample files:** The `Generic_ZipSync_Pentaho_Process_v7.zip` set of Pentaho configuration files that convert sample data into a ZipSync archive that can be used with EPIC. Use these files to explore how to use Pentaho to cleanse and format data for use with the EPIC tool, and then customize them for your own system.

## System requirements

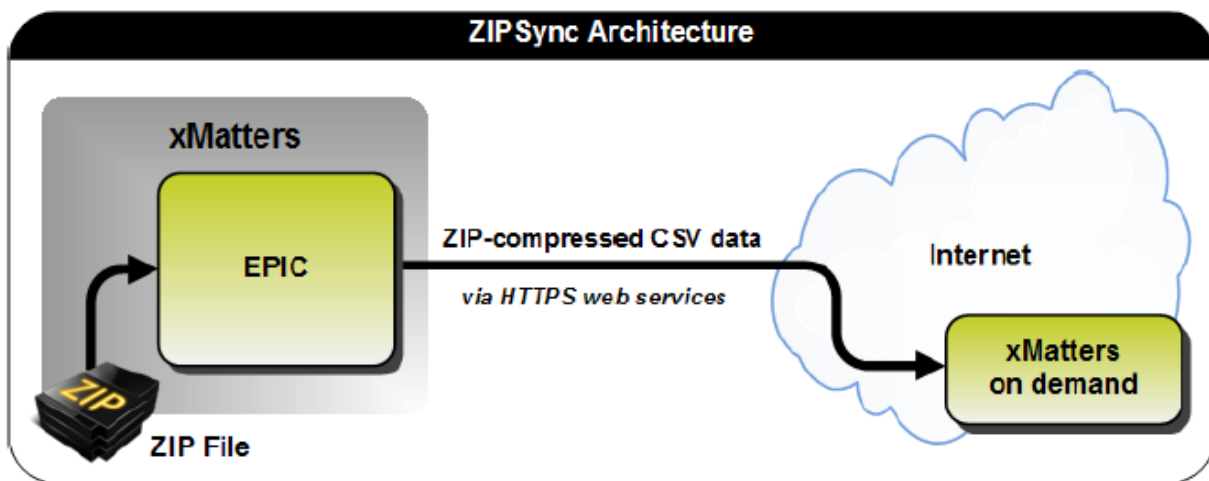
These examples have been tested on the Windows operating system. You may be able to run these examples on the Mac operating system if you modify the Pentaho configuration files to replace Windows batch files with the Mac equivalents and change other system variables accordingly.

Java version 6.0 must be installed on your system to run these components.

You must use an xMatters web service user account (not a standard user account) to authenticate with the EPIC tool. Also, your xMatters on demand deployment must contain a user with the user ID "epic" for the data import to be successful. If you are unsure of your web service account information or whether your deployment contains an epic user, consult your xMatters administrator.

## xMatters EPIC client overview

The EPIC data synchronization utility is a command-line tool that allows organizations to synchronize their user, device, and call-tree data with xMatters on demand. For example, an organization may want to synchronize data from its HR, LDAP, Active Directory, or other corporate system.



The EPIC tool's ZipSync mode allows you to synchronize data from *any* system into xMatters on demand. In ZipSync mode, data synchronization is done through a set of CSV (comma-separated value) files. You export data from your system to a predefined format in a set of CSV files, and then archive (or 'zip up') these files and synchronize them with xMatters on demand using a web service call.

### ZipSync mode file archive

The ZipSync archive file contains a set of CSV (comma-separated value) data files, and a manifest file. The manifest file (`manifest.xml`) allows you to specify the version of the ZipSync archive and configure synchronization options. The data files contain the data that is synchronized into xMatters on demand.

The following CSV data files are required to be in the ZIP archive:

- DS\_EMAIL\_DEVICES.csv
- DS\_FAX\_DEVICES.csv (ZipSync archive version 1.2 and later)
- DS\_GROUPS.csv
- DS\_PERSON\_SUPERVISORS.csv
- DS\_SITES.csv

- DS\_TEAM\_MEMBERSHIPS.csv
- DS\_TEAMS.csv
- DS\_TEXT\_PAGER\_DEVICES.csv
- DS\_TEXT\_PHONE\_DEVICES.csv
- DS\_USER\_CUSTOM\_VALUES.csv
- DS\_USER\_JOIN\_ATTRIBUTES.csv
- DS\_USERS.csv
- DS\_VOICE\_DEVICES.csv
- DS\_VOICE\_IVR\_DEVICES.csv (ZipSync archive version 1.1 and later)

For more information about the ZipSync archive file, see the *xMatters EPIC Data Synchronization guide* available as part of the EPIC client download at <http://community.xmatters.com/docs/DOC-3126>.

## Pentaho Data Integration overview

Pentaho Data Integration provides a full ETL solution that allows you to extract data from your system and convert it to the format required by the ZipSync archive files.

Pentaho Data Integration includes the following features:

- Rich graphical designer that empowers ETL developers
- Broad connectivity to any type of data, including diverse and big data
- Enterprise scalability and performance, including in-memory caching
- Big data integration, analytics, and reporting, including Hadoop, NoSQL, traditional OLTP & analytic databases
- Modern, open, standards-based architecture

Pentaho Data Integration's intuitive and rich graphical designer allows you to perform ETL quickly, without manual coding.

## Generic ZipSync/Pentaho sample files overview

The `Generic_ZipSync_Pentaho_Process_v7.zip` file includes sample Pentaho job and transformation files. If you do not have access to this file, contact your xMatters representative, or look for an attachment on the xMatters community page where you downloaded this document.

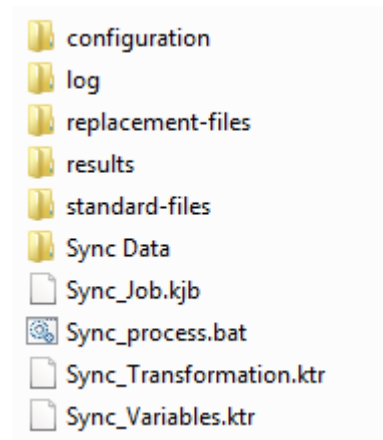
When Pentaho executes the job file it

- runs the Pentaho transformation files to convert the sample input data to an xMatters-friendly format.
- creates a ZipSync archive file that contains transformed input data.
- runs EPIC in ZipSync mode to import data into xMatters on demand.
- sends an email that contains the results of the ETL transformation in an Excel spreadsheet.

The Generic ZipSync/Pentaho sample files are designed to go to a pre-determined directory at a configured time interval and then process the data file it finds. This assumes that a new data file is put in place in this directory and overwrites an existing data file the same name.

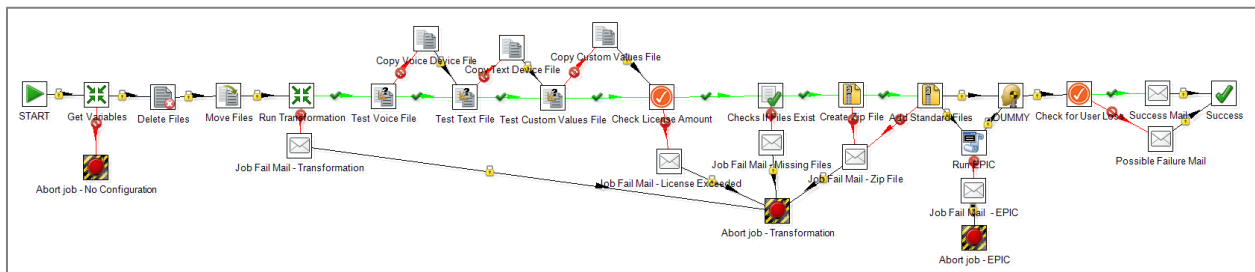
## Process files and directories

The Generic\_ZipSync\_Pentaho\_Process\_v7.zip archive file contains the following files and directories:



The top-level folder contains the job file, the batch file, and the transformation files:

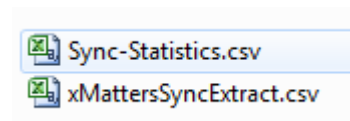
- **Sync\_Job.kjb:** the Pentaho job file. This is the job that you run to execute the data transformation.



- **Sync\_Process.bat:** a Windows batch file that runs the job file
- **Sync\_Variables.ktr:** the Pentaho transformation file used to pull in variables from the `Sync.properties` file
- **Sync\_Transformation.ktr:** the Pentaho transformation file used to convert the data in the single input CSV file into multiple ZipSync CSV files

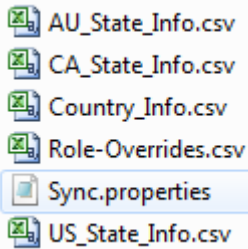
## Sync Data folder

This folder contains the sample data and statistics files;



## Configuration folder

This folder contains the properties, country, state, and role override files:



By default, users are imported into xMatters on demand with the role of No Access User. You can set specific roles for individual users by editing the `Role-Overrides.csv` file.

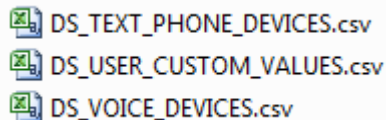
The `US_State_Info` (United States), `AU_State_Info` (Australia), `CA_State_Info` (Canada), and `Country_Info` files can be used to match names and codes to time zones.

### Log folder

This folder contains the transformation log and the Excel error spreadsheets produced by the Pentaho transformation. These files are included in the job summary email sent to the recipients configured in the `Sync.properties` file.

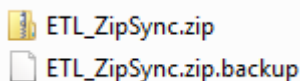
### Replacement-files folder

This folder contains some CSV files that are required be included in the ZipSync archive file. These files contain the required column headers but do not contain any data. These files are copied to the results directory if the corresponding files produced by the data transformation are empty.



### Results folder

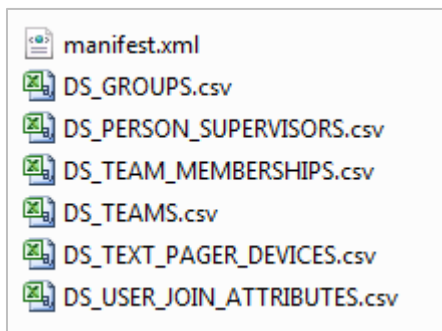
This folder contains the ZipSync archive file that is imported into xMatters on demand with the EPIC client.



### Standard-files folder

This folder contains the `manifest.xml` and some CSV files that are required by the ZipSync archive file but are not produced as part of the Pentaho ETL transformation process. The CSV files contain the required headers but do not contain any data.





## Generic ZipSync configuration

Follow the steps below to set up the environment.

### About this example

This example shows how to use the provided Pentaho configuration files to convert a sample data into a ZipSync archive file, validate the archive file with EPIC, and send a summary email. Running these configuration files doesn't import the data into your xMatters on demand deployment, because the EPIC command is run in validate-only mode. However, you can modify the Pentaho configuration files to run EPIC in a mode that imports data into xMatters.

### Download and configure the EPIC client

Download the version of the EPIC client that corresponds to your version of xMatters on demand and configure it to connect to your xMatters on demand instance.

For example, if you are using version 5.5.54 of xMatters on demand, download `epic-client-5.5.54.zip`.

You can download the EPIC client from the xMatters community site, <http://community.xmatters.com>.

1. Go to **Community > Quick Links > Downloads > xMatters on demand**, or navigate to the following link: <http://community.xmatters.com/docs/DOC-3126>.
2. Click the **Download** link to download the version of EPIC that corresponds to your version of xMatters on demand.
3. Unzip the EPIC client file to a location of your choice. The following examples assume that you unzip this file to `C:\epic-client-5.5.54\`.
4. Edit the `epic-client-5.5.54\conf\transport.properties` file and configure the authentication settings that allow EPIC to log on to your xMatters on demand deployment:

<b>WS_URL</b>	The URL of your xMatters on demand deployment. Typical values include the following: <ul style="list-style-type: none"><li>• <a href="https://na1.xmatters.com">https://na1.xmatters.com</a></li><li>• <a href="https://eu1.xmatters.com">https://eu1.xmatters.com</a></li><li>• <a href="https://au1.xmatters.com.au">https://au1.xmatters.com.au</a></li><li>• <a href="https://Generic.hosted.xmatters.com">https://Generic.hosted.xmatters.com</a></li></ul>
---------------	--

<b>COMPANY_NAME</b>	Your xMatters on demand company name.
<b>WS_USERNAME</b>	The user name of a web service user.
<b>WS_PASSWORD</b>	The password of the web service user. <b>Note:</b> Contact your xMatters on demand administrator for a web service user name and password.

5. If your system uses a proxy server, update the following values:

<b>PROXY_SERVER_NAME</b>	Proxy address
<b>PROXY_PORT</b>	Proxy port
<b>PROXY_NTLM_NAME</b> (EPIC client version 5.5.54 and later.)	Address for NTLM proxy servers. This value is not required for non-NTLM proxy servers.

6. Save the `transport.properties` file.
7. Test the authentication settings in the `transport.properties` file by running EPIC with a non-existent data file. Because this data file does not exist, running EPIC returns an error that states that the file could not be found. Open a command prompt, and navigate to the directory where you installed EPIC.
8. From the `epic-client-5.5.54\bin` folder, run the following command:  
**epic file nothing.zip**

If the authentication settings are configured correctly, you receive an error that the data file does not exist, but you do not receive an authentication error. However, if the authentication settings are configured incorrectly, you receive both an authentication error and an error that the data file does not exist.

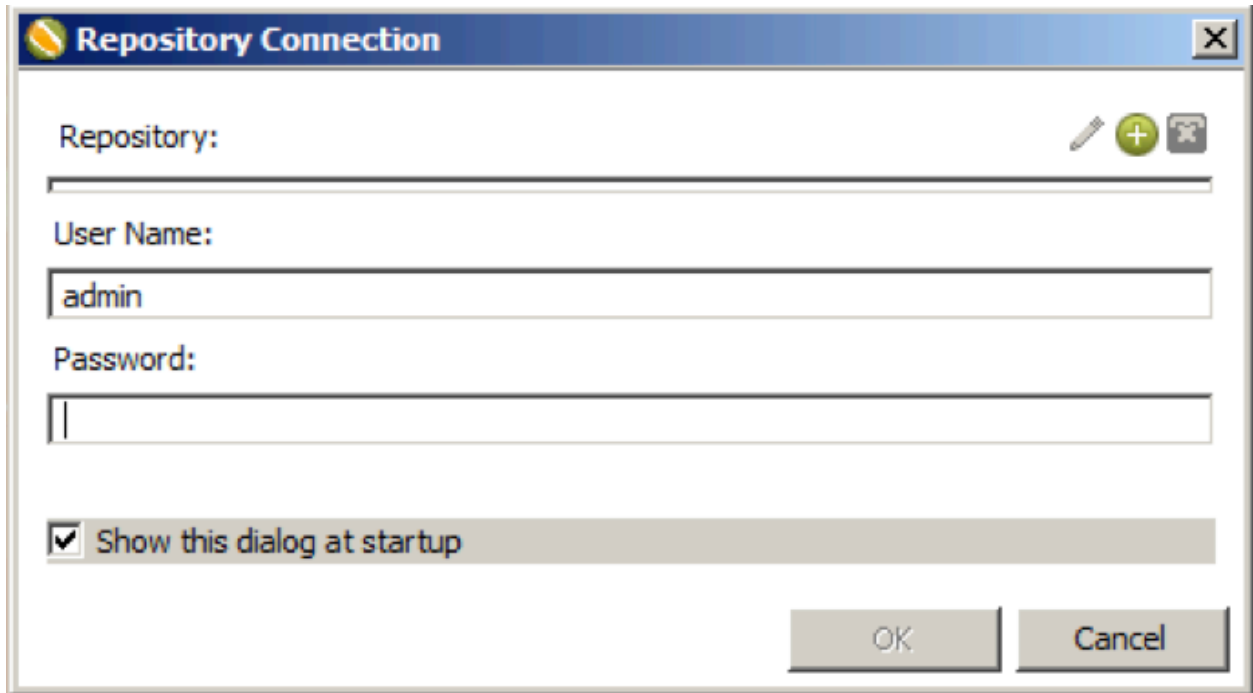
**Note:** If a Java Runtime error occurs, you need to install the Java Runtime Environment (JRE).

## Download and install Pentaho

Download and install Pentaho so that you can use it to run the sample ETL files. These instructions assume that you are running Pentaho on Windows.

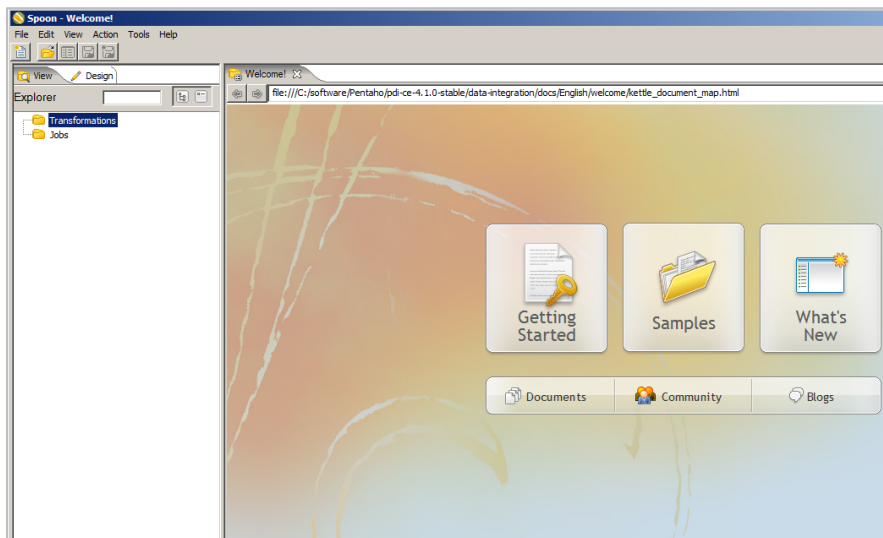
1. Download **pdi-ce-5.0.1-stable.zip** from the following location:  
<http://sourceforge.net/projects/pentaho/files/Data%20Integration/5.0.1-stable/>
2. Unzip the file to a directory on your server. The following examples assume that you unzip this folder to `C:\data-integration\`

3. Open a command prompt, and run **data-integration\Spoon.bat**.  
The Repository Connection screen appears.



4. Click **Cancel**.
5. If a Tip window appears, click **Close**.

The main Pentaho screen is displayed.



## Configure the Generic ZipSync support files

1. Unzip the `Generic_ZipSync_Pentaho_Process.zip` file to a folder on the same system where Pentaho is deployed. If you do not have access to this file, contact your xMatters representative, or look for an attachment on the xMatters community page where you downloaded this document. The following examples assume that you have unzipped this file to `c:\Generic_ZipSync_Pentaho_Process`.
2. Edit the variables in the `configuration\Sync.properties` to define the directory locations, mail settings, and other properties required by the Pentaho job file.

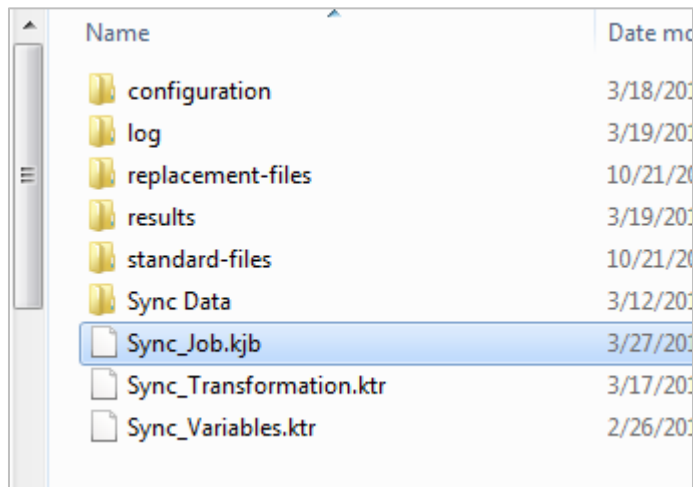
**Note:** Paths must contain forward slashes (/) instead of backward slashes (\).

Variable	Description	Example
<b>base_directory</b>	Directory that contains the sample ETL files.	<code>c:/Generic_ZipSync_Pentaho_Process</code>
<b>source_directory</b>	Directory that contains the sample data file.	<code>c:/Generic_ZipSync_Pentaho_Process/Sync Data</code>
<b>epic_directory</b>	Directory of the EPIC utility.	<code>c:/epic-client-5.5.54</code>
<b>Log_File</b>	Transformation log file location.	<code>c:/Generic_ZipSync_Pentaho_Process/log/Sync_Transformation</code>
<b>file_name</b>	Name of the ZipSync archive file. Pentaho creates this file in the results directory.	<code>ETL_ZipSync.zip</code>
<b>mail.sender</b>	Email address of an account for job status emails.	<code>me@mycompany.com</code>
<b>mail.destination</b>	A list of mail addresses to send job status messages to. Use a space between addresses.	<code>me@mycompany.com you@yourcompany.com</code>
<b>mail.port</b>	Port to use for sending mail.	This value is usually 25.
<b>mail.authentication.user</b>	User ID of email sender.	<code>me@mycompany.com</code>
<b>mail.authentication.password</b>	Password of email sender.	*****
<b>user_loss_threshold</b>	Change in the number of users that flags a warning. If more than the specified number of users changes during the data synchronization process, a warning is generated.	5
<b>device_delay_1</b>	The email device delay, in minutes. (integer value)	0
<b>device_order_1</b>	Integer sequence number for the order of the email device in the list of all users' devices. The sequence begins with 0.	0
<b>device_delay_2</b>	The work phone device delay, in minutes. (integer value)	1
<b>device_order_2</b>	Integer sequence number for the order of the work phone device in the list of all users' devices. The sequence begins with 0.	1
<b>device_delay_3</b>	The cell phone device delay, in minutes. (integer value)	2
<b>device_order_3</b>	Integer sequence number for the order of the cell phone device in the list of all users' devices.	2

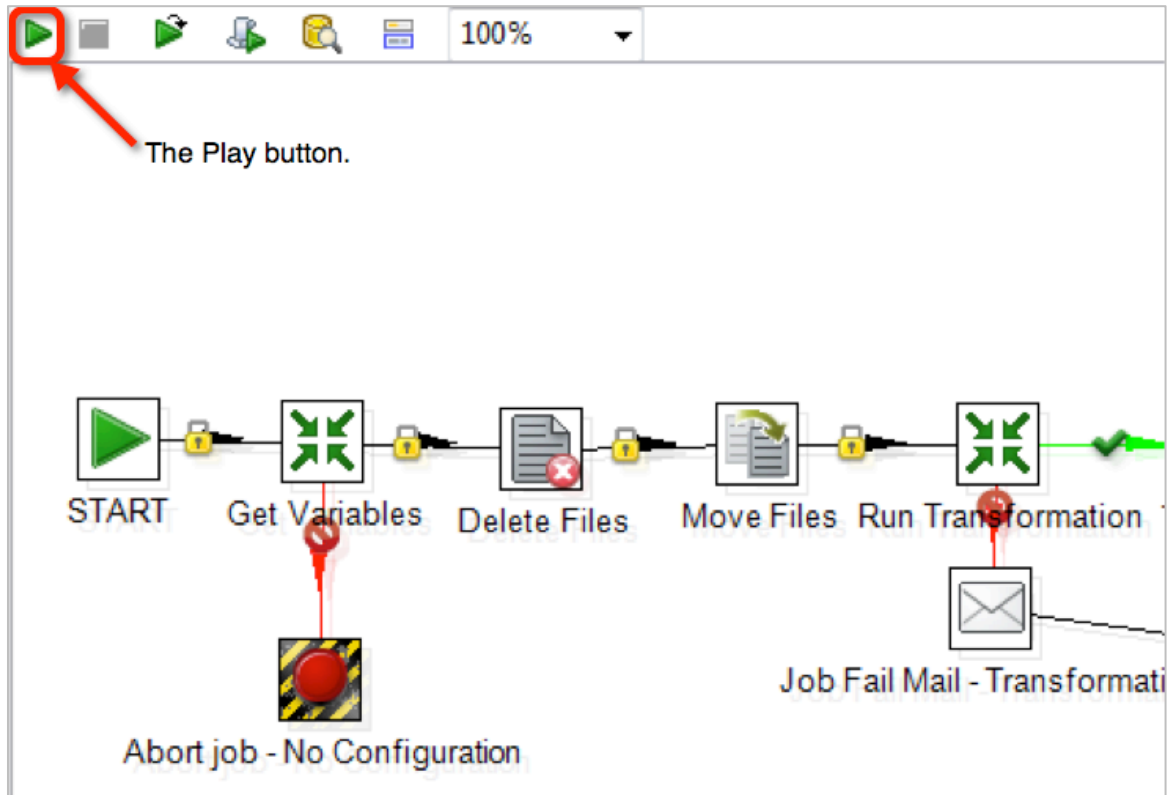
	The sequence begins with 0.	
<b>device_delay_4</b>	The SMS device delay, in minutes. (integer value)	3
<b>device_order_4</b>	Integer sequence number for the order of the SMS device in the list of all users' devices. The sequence begins with 0.	3
<b>licensed_users</b>	Number of licensed users for the xMatters instance.	Verify the number of xMatters licenses that is allowed for your deployment.

## Run the job

1. From Pentaho, select **File > Open**, and open the Pentaho job file `Sync_Job.kjb` located in the `Generic_ZipSync_Pentaho_Process` folder.



2. Click the green **Play** button to run the Pentaho job.



## Customize the transformation files

Running the provided transformation files as they are supplied does not import the sample into your xMatters system because the EPIC client is configured to run in validation-only mode. You can customize the transformation files to work with data from your system and commit the transformed data into xMatters.

### Configure mirror mode or non-mirror mode

When EPIC is run in ZipSync mode with the default settings, updates are performed in mirror mode. When an update is performed in mirror mode, records that are not included in the synchronization archive file (and are marked as externally owned) are deleted from the xMatters instance.

To avoid accidentally deleting existing data, you may want to run EPIC in non-mirror mode while you fine-tune the data import process. The sample Pentaho transformation files use EPIC in mirror mode unless you explicitly turn it off.

To turn off mirror mode, open the `manifest.xml` file of the ZipSync archive, and change the value of the "mirror" Key to N.

```
<entry key="mirror">N</entry>
```

## External ownership

The configuration files in this example mark data as externally owned. Externally-owned data can only be deleted by running another EPIC process, and cannot be deleted manually through the xMatters user interface, even by a power user. If you want to be able to delete users from the xMatters user interface, set the IS\_EXTERNALLY\_OWNED flag of each field to N.

To change the IS\_EXTERNALLY\_OWNED flag, perform the following steps:

1. In Pentaho, open the transform file, `Sync_Transformation.ktr`.
2. Double-click the “Add Constants” step for the item you want to mark as not externally owned. For example, double-click the “Add User Constants” to open the file that sets constants for users in the `DS_USERS.csv` file.
3. Set the Value column of the IS\_EXTERNALLY\_OWNED row to N.

## Use your own data instead of sample data

The sample transformation files convert the data in the `xMattersSyncExtract.csv` file to a ZipSync archive file. You will need to edit the Pentaho files to use your system’s data as source data instead of the provided sample data. This requires working knowledge of the Pentaho system and your system.

## Configure EPIC to commit changes

Once you have fine-tuned the process and are ready to commit your changes to xMatters, remove the validate-only (`-v`) flag from the EPIC command.

1. Open the `Sync_Job.kjb` file.
2. Double-click the **Run Epic** step.
3. In the Script File Name field, remove the `-v` option from `${epic_directory}/bin/epic.bat -v`

After you run EPIC in commit mode, you can log on to the xMatters on demand user interface and view the synchronization report to view the results of the data synchronization process and troubleshoot any errors. For more information about interpreting synchronization reports, see the *xMatters EPIC Data Synchronization guide*, available as part of the EPIC client download at <http://community.xmatters.com/docs/DOC-3126>.

## ETL flow

The Generic ZipSync ETL process builds the CSV files that are most commonly required to synchronize Human Resources (HR) data into xMatters:

- `DS_EMAIL_DEVICES.csv`
- `DS_SITES.csv`
- `DS_TEXT_PHONE_DEVICES.csv`
- `DS_USER_CUSTOM_VALUES.csv`
- `DS_USERS.csv`
- `DS_VOICE_DEVICES.csv`

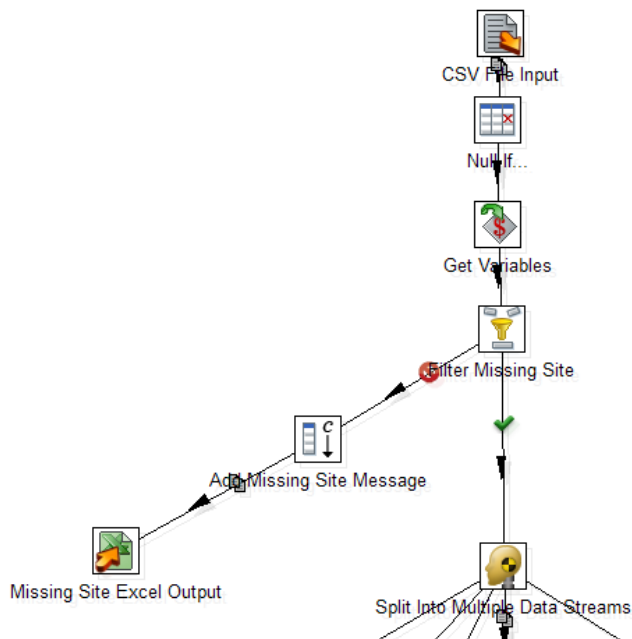
The files are built as part of the data transformation done by the Sync\_Transformation process (Sync\_Transformation.ktr).

This process has 7 stages:

1. Clean and validate input data
2. Build the sites file
3. Build the users file
4. Build the email devices file
5. Build the voice devices and text phone devices files
6. Build the custom values file
7. Gather the statistics from this job run and compare it to the last run

Each stage is discussed in more detail in the following sections.

## 1. Clean and validate input data

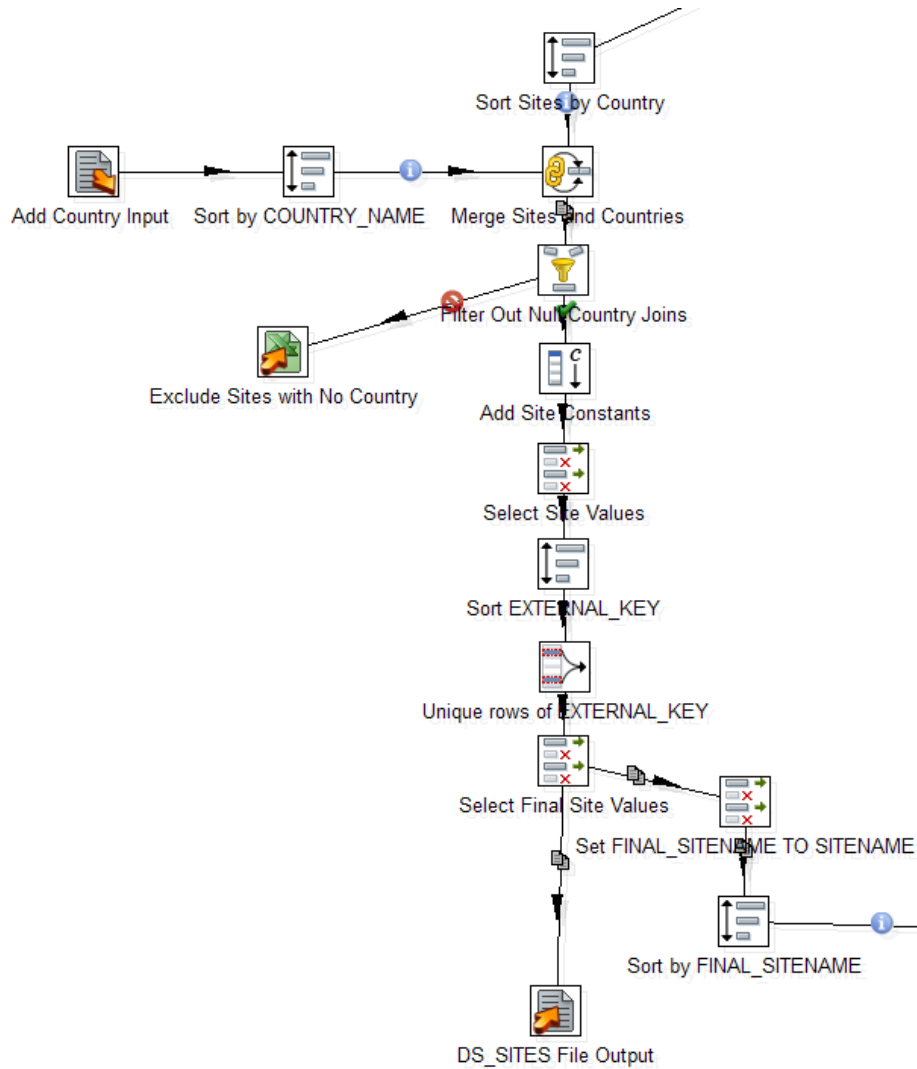


### Workflow

1. The data is mapped in from a CSV or Excel file.
2. All values are set to NULL if they are empty strings.
3. Get the variables from the parent job.
4. Filter out rows that do not have a site value and create a spreadsheet of bad data.
5. Split the data into multiple streams of data transformation.



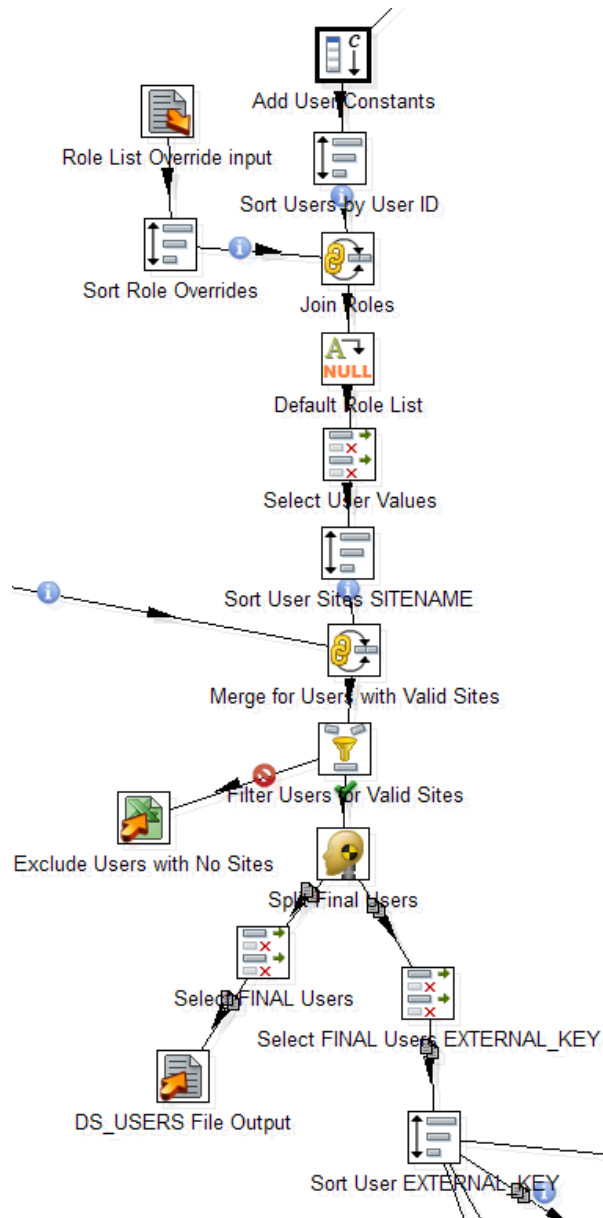
## 2. Build the sites file



### Workflow

1. Sort the sites by the Country field.
2. Join the data from the input file to the Country\_Info file to get the associated time zones.
3. Filter out rows that do not have a country value from the Country\_Info file and create a spreadsheet of bad data.
4. Add constants that are required for the DS\_SITES.csv file but are not supplied by the input file.
5. Select and rename data.
6. Sort the data by the site name and then remove any duplicate rows.
7. Based on the final values, create the DS\_SITES.csv file and a list of site names to be compared with user data.

## 3. Build the users file

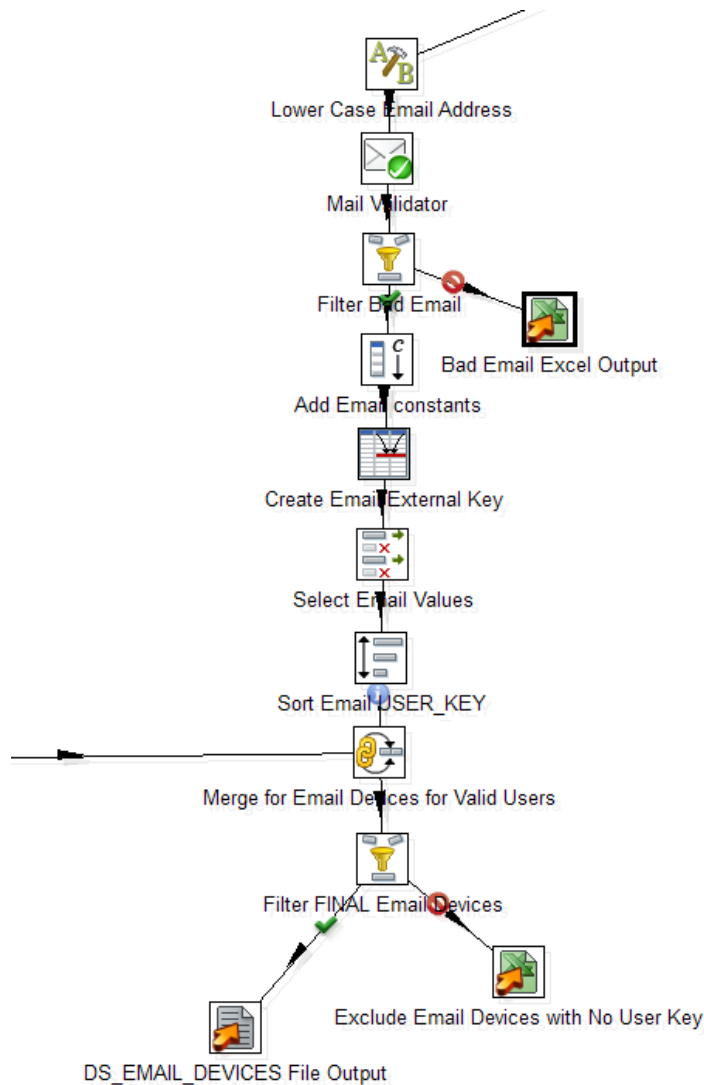


### Workflow

1. Add constants that are required for the DS\_USERS.csv file but are not supplied by the input file.
2. Sort the sites by the user field.
3. Join the data from the input file to the role list overrides file to get the associated roles for specific users.
4. Add the default role if the role list value is still null.
5. Select and rename data and sort by site name.

- Join with the final site list and then filter out rows that do not have a site match and create a spreadsheet of bad data.
- Based on the final values, create the `DS_ USERS.csv` file and a list of user IDs to be compared with device and custom value data.

## 4. Build the email devices file

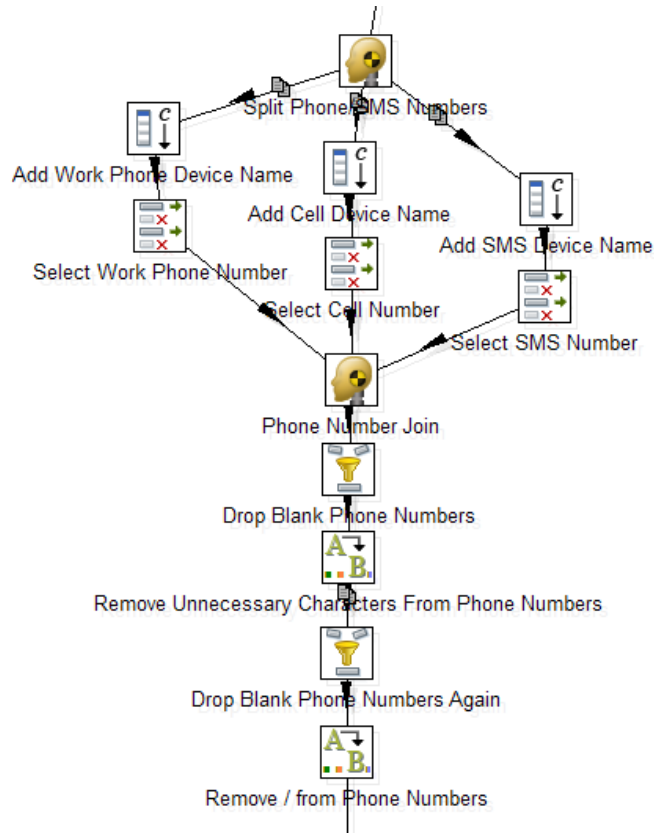


### Workflow

1. Reformat the email addresses into lower-case alphabetic characters.
2. Validate email and filter out bad email addresses to a spreadsheet.
3. Add constants that are required for the `DS_EMAIL_DEVICES.csv` file but are not supplied by the input file.
4. Create the external key for the row by combining the user ID and the device name.
5. Select and rename data and sort by user ID.

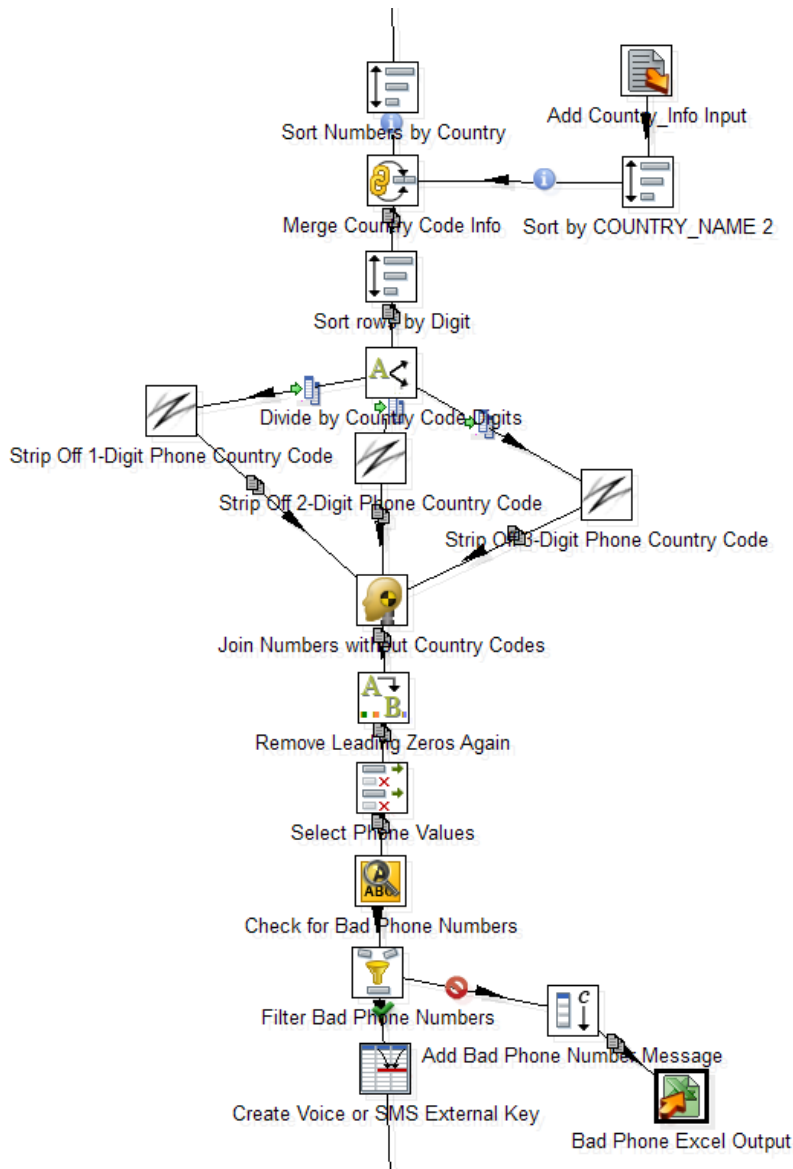
6. Join with the final user list and then filter out rows that do not have a user match and create a spreadsheet of bad data.
7. Based on the final values, create the DS\_EMAIL\_DEVICES.csv file.

## 5. Build the voice and text devices files

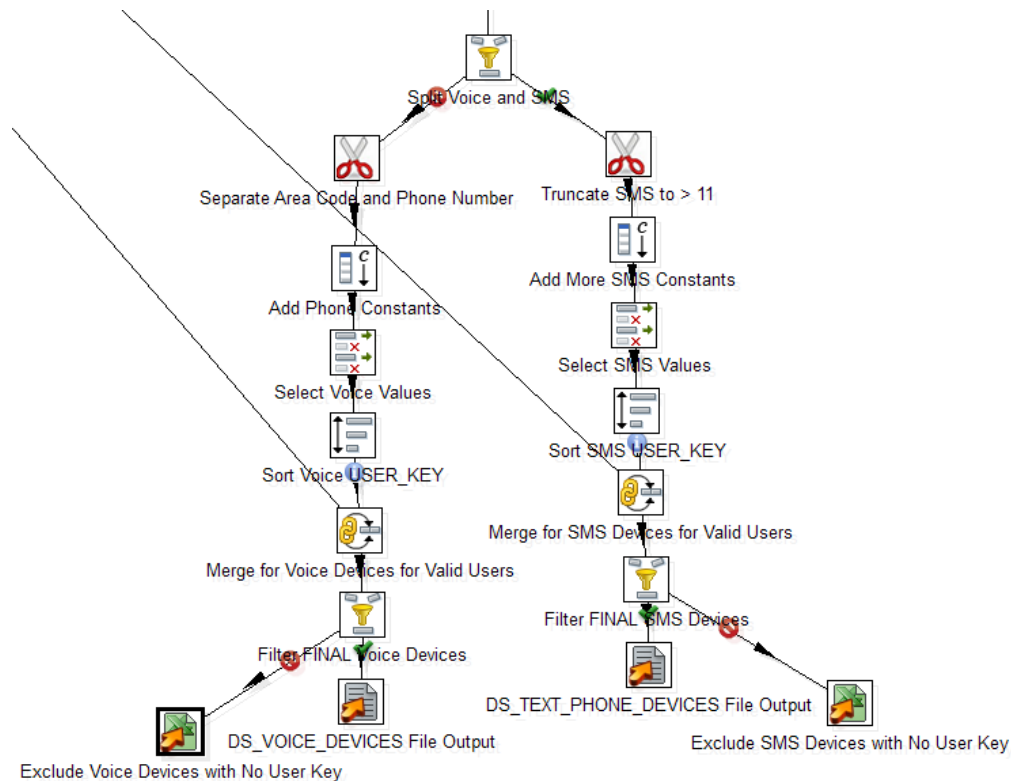


### Workflow

1. Split the data into multiple streams of device data transformation.
2. Add constants that are required for the DS\_VOICE\_DEVICES.csv and DS\_TEXT\_PHONE\_DEVICES.csv files but are not supplied by the input file.
3. Select and rename data.
4. After joining the different device flows, drop any blank phone numbers.
5. Remove unnecessary characters in phone numbers.



- Sort phone numbers by Country.
- Join the data from the input file to the Country\_Info file to get the associated time zones.
- Sort the data by the number of digits and divide the data.
- Strip off the country code if present.
- Rejoin streams and remove any leading zeros.
- Select and rename data.
- Test for bad phone numbers and filter out the bad data to a spreadsheet.
- Create the external key for the row by combining the user ID and the device name.



14. Split the voice devices from the SMS devices.

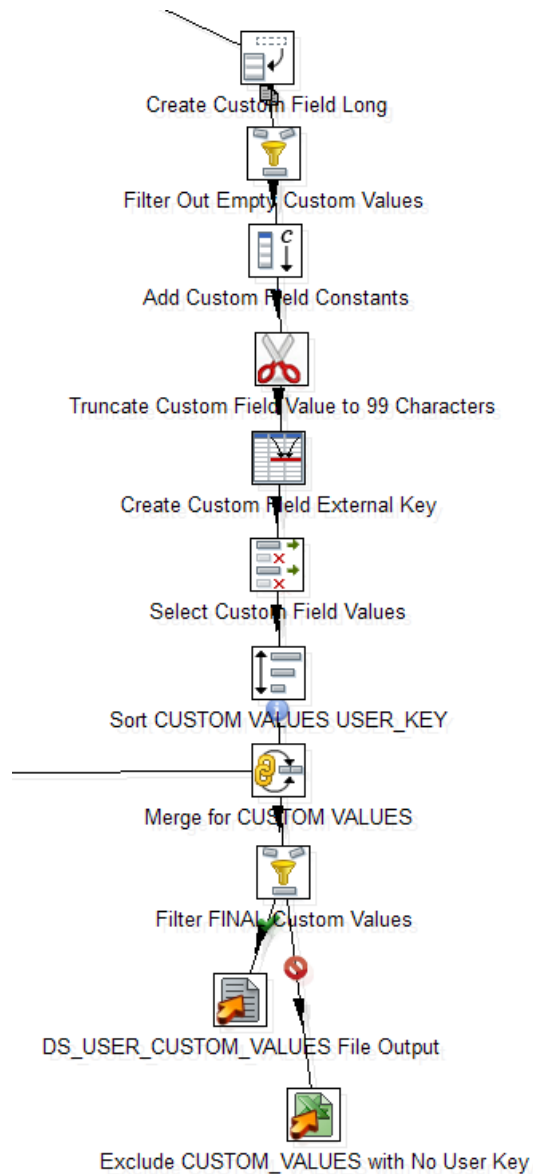
15. For the voice devices:

- Separate the area code from the phone number.
- Add constants that are required for the `DS_VOICE_DEVICES.csv` file but are not supplied by the input file.
- Select and rename data and sort by user ID.
- Join with the final user list and then filter out rows that do not have a user match and create a spreadsheet of bad data.
- Based on the final values, create the `DS_VOICE_DEVICES.csv` file.

16. For the SMS devices:

- Truncate SMS phone numbers to less than 11 digits.
- Add constants that are required for the `DS_TEXT_PHONE_DEVICES.csv` file but are not supplied by the input file.
- Select and rename data and sort by user ID.
- Join with the final user list and then filter out rows that do not have a user match and create a spreadsheet of bad data.
- Based on the final values, create the `DS_TEXT_PHONE_DEVICES.csv` file.

## 6. Build the custom values file

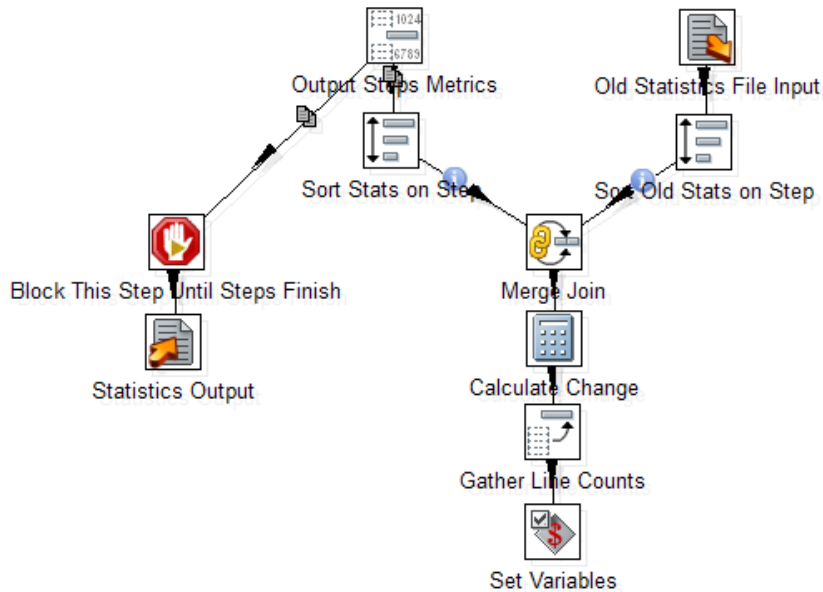


### Workflow

1. De-normalize the custom value data and filter out blank values.
2. Add constants that are required for the `DS_USER_CUSTOM_VALUES.csv` file but are not supplied by the input file.
3. Truncate the custom field values to be fewer than 100 characters.
4. Create the external key for the row by combining the user ID and the custom value name.
5. Select and rename data and sort by user ID.

- Join with the final user list and then filter out rows that do not have a user match and create a spreadsheet of bad data.
- Based on the final values, create the `DS_USER_CUSTOM_VALUES.csv` file.

## 7. Gather the statistics from this job run and compare it to the last run



### Workflow

- Read in the statistics file from the last transformation run and sort based on step.
- Create output metrics based on transformation steps and sort them based on step.
- Merge the two data streams and calculate the differences.
- Gather line counts and set the variables.
- Wait until all of the steps in the transformation complete and then write the current statistics to file.

## Generic ZipSync customization

Before you customize the Generic ETL process for your own system, you should deploy and test this process with the sample data to ensure that the process runs without errors.

## Understanding the job status emails

At the end of processing, Pentaho generates an email and sends it to the recipient or distribution list that is configured in the `Sync.properties` file. Ideally, these emails should go to a person or team who can contact HR or specific users about errors in their contact information. The body of the email includes a summary of any problems with the data and compares this job to the previous one.

The following screenshots show an example of an output email:



## Completion Report:

### Data With Problems:

Records with missing email addresses:	0
Records with bad email addresses:	0
Records with bad phone numbers:	0

### Comparison Information

Synchronization Type	Current Count	Previous Count	Difference
Successful sites:	4	4	0
Successful users:	5	5	0
Successful email devices:	5	5	0
Successful SMS devices:	3	3	0
Successful voice devices:	0	0	0
Successful custom fields:	0	0	0

The following files may be attached to the email:

- Sync\_Errors\_Bad\_Email\_Addresses\_<date>.xls
- Sync\_Errors\_Bad\_Phone\_Numbers\_<date>.xls
- Sync\_Errors\_Custom\_Values\_Devices\_No\_User\_Key\_<date>.xls
- Sync\_Errors\_Email\_Devices\_No\_User\_Key\_<date>.xls
- Sync\_Errors\_Missing\_Site\_<date>.xls
- Sync\_Errors\_Missing\_Work\_Email\_<date>.xls
- Sync\_Errors\_SMS\_Devices\_No\_User\_Key\_<date>.xls
- Sync\_Errors\_Users\_No\_Sites\_<date>.xls
- Sync\_Errors\_Voice\_Devices\_No\_User\_Key\_<date>.xls
- Sync\_Transformation\_<date>.log
- Sync-Statistics.csv

These files are not added if no errors occur.

The following screenshot shows an example of an error in the Sync\_Errors\_Bad\_Phone\_Numbers file:

User	Country	DEVICE_NAME	DEVICE_ORDEVICE_DE	java_phone_number	ERROR_MESSAGE
45171	Canada	TEXT_PHONE_NUMBER	0	0	440783E12 Invalid Phone number

In this situation, the user is loaded but the invalid phone number is not loaded. You can then cross reference this data with the original input file to see the number that was listed there, which in this example is 407830000000.

You can use the `user_loss_threshold` property in the `sync.properties` file to flag when there is a large change in the number of users that are loaded in the system. This can alert you if a large number of users are suddenly deleted from the system, which could indicate an error. The following screenshot shows an example of an email that is flagging a change in the number users that is greater than the threshold:

Subject: Possible Job Failure: BCP Job Complete

Message

- BCM\_Errors\_Custom\_Values\_Devices\_No\_User\_Key\_20121109.xls (14 KB)
- BCM\_Errors\_Bad\_Phone\_Numbers\_20121109.xls (84 KB)
- BCM\_Errors\_SMS\_Devices\_No\_User\_Key\_20121109.xls (14 KB)
- BCM-Statistics.csv (1 KB)
- BCM\_Errors\_Voice\_Devices\_No\_User\_Key\_20121109.xls (14 KB)
- BCM\_Errors\_Missing\_Work\_Email\_20121109.xls (14 KB)

### Completion Report:

**Data With Problems:**

Records with missing email addresses:	0
Records with bad email addresses:	0
Records with bad phone numbers:	395

## Comparison Information

Synchronization Type	Current Count	Previous Count	Difference
Successful sites:	176	242	-66
Successful users:	2519	9311	-6792
Successful email devices:	2519	9310	-6791
Successful SMS devices:	977	1932	-955
Successful voice devices:	3293	8185	-4892
Successful custom fields:	20766	68853	-48087