PUBLIC

2017-05

SAP Ariba integration toolkit guide

SAP Ariba cloud solutions

SAP

SAP Ariba

Content

The SAP Ariba integration toolkit

In this section:

SAP Ariba integration toolkit overview

The SAP Ariba integration toolkit allows buying organizations to integrate their ERP systems with an SAP Ariba cloud solution to exchange master and transactional data via CSV file upload and download.

The SAP Ariba integration toolkit is made up of two tools:

- The **SAP Ariba data transfer tool**, a command-line utility that facilitates CSV file transfer in batch mode between your SAP Ariba system and your back-end system, manually-maintained data, or some other system.
- The **DB Connector**, a command-line utility that allows you to extract master data and transactional data from JDBC-based back-end system database tables and save it to CSV files or upload data to the ERP system.

The SAP Ariba data transfer tool includes additional functionality if being used in an SAP environment. For information about using the tool in an SAP environment, see the *Integrating SAP Ariba cloud solutions procurement and invoicing data with SAP ERP and SAP S/4HANA*.

The following graphic provides a high-level overview of how data flows between your SAP Ariba system and a back-end system through the DB Connector and the SAP Ariba data transfer tool:

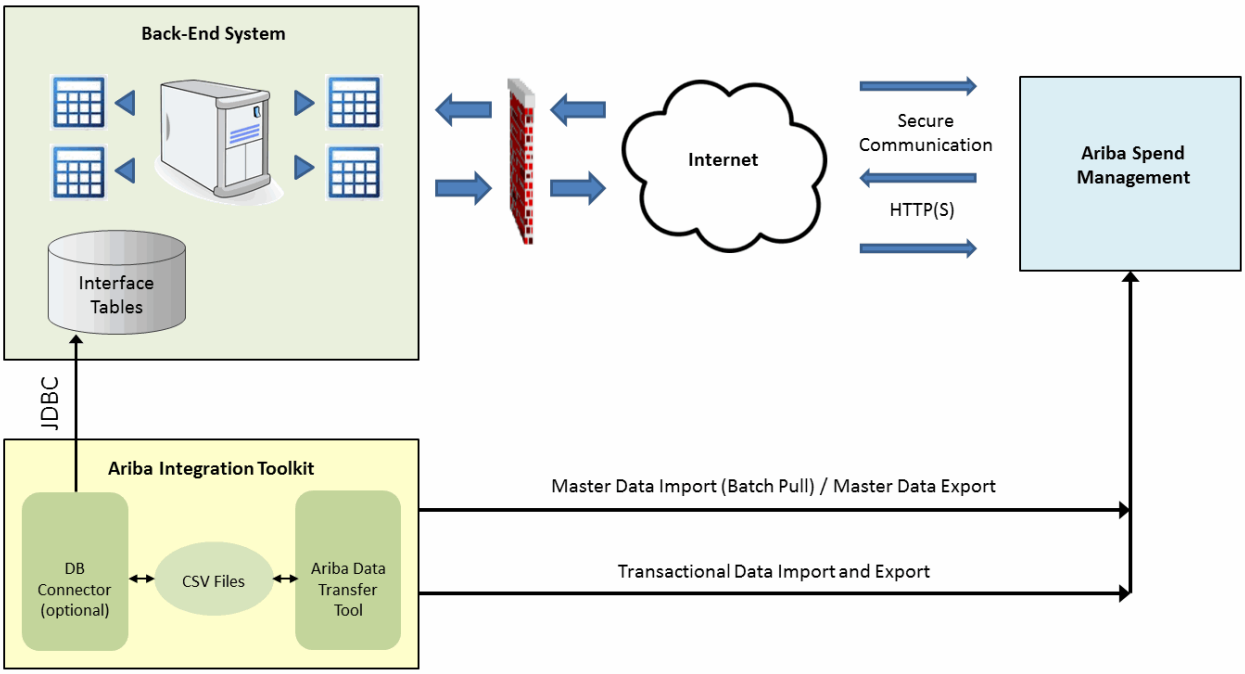


Figure 1: Data Flow Between Your SAP Ariba System and a Back-End System

This guide is useful for users who configure the SAP Ariba integration toolkit. You are expected to know how to run scripts on the command line in either UNIX or Microsoft Windows. You must also understand environment variables and know how to edit scripts and batch files.

About the SAP Ariba data transfer tool

The SAP Ariba data transfer tool is a command-line utility that facilitates data transfer in batch mode between your back-end system and your SAP Ariba system. Data is transferred in the form of comma-separated-value (CSV/XLS) files. You can use the SAP Ariba data transfer tool with a scheduler to export transactional data, such as expense reports, from SAP Ariba to your back-end system or import or export master data from your back-end system to SAP Ariba. You can import master data only as a batch import task.

The SAP Ariba data transfer tool is a simple HTTP client that resides on a system behind your firewall and runs on Microsoft Windows or UNIX. You control the SAP Ariba data transfer tool by setting operational parameters in separate options files.

Using the SAP Ariba data transfer tool

Procedure

1. Download and install the SAP Ariba integration toolkit. See Installing or Upgrading the Integration Toolkit [page 6].
2. Encrypt your shared secret. See About Security Encryption [page 24].
3. Edit and save an options file for each task. See Customizing the Sample Options Files [page 26].
4. Run the main program specifying the options file as an argument. See Ariba Data Transfer Tool Syntax [page 47].

About the DB Connector

The DB Connector is an optional command-line utility that bridges the gap between the SAP Ariba data transfer tool and your JDBC-based back-end system. It allows you to extract master data and transactional data from back-end system database tables and save it to CSV files. It also imports transactional data exported from the Ariba system (CSV files) into the back-end system database tables. For example:

The DB Connector extracts master data from your back-end system that is then uploaded to your Ariba system by the SAP Ariba data transfer tool.

The DB Connector uploads transactional data, such as receipts and payment requests, exported from the Ariba system by the SAP Ariba data transfer tool to the back-end system.

If your back-end system is SAP, do not use the DB Connector. Instead, use the SAP transports described in the *Integrating SAP Ariba cloud solutions procurement and invoicing data with SAP ERP and SAP S/4HANA*.

Using the DB Connector

Procedure

1. Download and install the SAP Ariba integration toolkit. See Installing or Upgrading the Integration Toolkit [page 6].
2. Encrypt your database password. See About Encrypting The Database Password [page 74].
3. Establish directory layouts for the scenarios you plan to run. See Creating Directory Structures [page 76].
4. Create properties files for the transactions you plan to run. See Creating Properties Files [page 77].
5. Create query files for the data you plan to extract and import. See Creating SQL Query Files [page 81].
6. Run the tool. See Running the DB Connector [page 85].

Scheduling the Tools

There are two ways to schedule the running of tools in the SAP Ariba integration toolkit:

- The SAP Ariba data transfer tool and the DB Connector can be independently scheduled to run by an external scheduler.
- The SAP Ariba data transfer tool and the DB Connector can be run serially by a single script.

Setting up the SAP Ariba integration toolkit

In this section:

Installing or upgrading the integration toolkit

In this section:

Upgrading the integration toolkit

Context

If you are using an older version of the SAP Ariba integration toolkit, use the following procedure to upgrade the toolkit. If you are not sure about which version you have installed, see Using the -version and -info Options [page 16].

Procedure

1. Go to connect.ariba.com and log in. If you do not have a User ID and Password for Connect, contact your SAP Ariba account executive.
2. In the **Product Summary** portlet, click the Product Info page for your SAP Ariba solution.
3. In the **Administrator Tools** section, click **Ariba Integration Toolkit**.
4. Click **Download** and save the `aribaintegrationtoolkit.jar` file to your local system.
5. Extract the contents of `aribaintegrationtoolkit.jar` using the following command:

On Windows: `java -jar aribaintegrationtoolkit.jar <new directory>`.

On Unix: `java -jar aribaintegrationtoolkit.jar /<new directory>`.

> **Note**
>
> Make sure you are extracting the files to a location different from your existing installation of the SAP Ariba integration toolkit, such as `IntegrationToolkit_11s2`.

6. Copy all your .bat and .sh files available in different folders under the `sample-options\DataTransferTool` folder in your existing installation and replace these folders under the `sample-options\DataTransferTool` folder available in your newly created folder structure.
7. If you are upgrading from version 10s1 or older, edit the `sample-options` file corresponding to each of the .bat and .sh files available in your newly created folder structure with the following information:
   - Remove the `sharedsecret` parameter from the sample-options file.
   - Add three parameters, `integrationPasswordKeyPath`, `encryptionKeyPath`, and `servicemode` to the sample-options file if they are not already included in the file.
   - In the sample-options file, set `JAVA_HOME` to point to the location where you have installed the JRE.

Installing the SAP Ariba integration toolkit

Prerequisites

Before installing the SAP Ariba integration toolkit, complete the following prerequisites:

- If you are using the JDBC diver, install the appropriate JDBC database driver and write down its location and class. Supported drivers include:
  - Oracle JDBC type 4 driver
  - DB2 JDBC type 4 Universal driver
  - Microsoft SQL Server 2005 type 4 JDBC Driver 1.1
- Install Java JRE 1.6 or higher on your system.

Procedure

1. Go to connect.ariba.com and log in. If you do not have a User ID and Password for Connect, contact your SAP Ariba account executive.
2. In the **Product Summary** portlet, click the Product Info page for your SAP Ariba solution.
3. In the **Administrator Tools** section, click **Ariba Integration Toolkit**.
4. Click **Download** and save the `aribaintegrationtoolkit.jar` file to your local system.
5. Extract the contents of `aribaintegrationtoolkit.jar` using the following command:

   On Windows: `java -jar aribaintegrationtoolkit.jar <new directory>`.

   On Unix: `java -jar aribaintegrationtoolkit.jar /<new directory>`.

The command extracts the contents of `aribaintegrationtoolkit.jar` into the `<new directory>`.

Next Steps

After you have downloaded and extracted the SAP Ariba integration toolkit, proceed with the following steps:

- Configure security. See Configuring integration toolkit security [page 13].
- Configure the Ariba Transfer Tool. See Using the SAP Ariba data transfer tool [page 21].
- Configure the DB Connector. See Using the DB Connector [page 70].

SAP Ariba integration toolkit directories

When you extract the SAP Ariba integration toolkit jar file, the following directories are created. Files with the `.bat` extension are for use with Microsoft Windows. Files with the `.sh` extension are for use with UNIX.

Directory	Contents
`bin`	Wrapper scripts for the SAP Ariba data transfer tool and the DB Connector: • `aribafiletransfer.*` • `downloadutils.*` • `encryptstring.*` • `toolslib.*` • `uploadutils.*` • `encryptcustomerkey.*`
`classes`	The `log4j.properties` file.
sample-bin	Script files used by the DB Connector: • `dbctool_db2_export.*` • dbctool_db2_import.* • dbctool_ora_export.* • dbctool_ora_import.*

Directory	Contents
`sample-options`	Customizable sample options files for data uploads, data downloads, and SAP Ariba Spend Analysis: In `DataTransferTool\data-download`: • `transactionaldata.*.sample` • masterdata.*.sample • `wsdata.bat.sample (Windows) for export of contract workspaces` • `wsdata.sh.sample (Unix) for export of contract workspaces` In `DataTransferTool\data-upload`: • `incremental-masterdata.*.sample` • `masterdata.*.sample` • `transactionaldata.*.sample` In `DataTransferTool\spend-visibility`: • `upload.*.sample` In `DataTransferTool\sv-download`: • `download.*.sample` In `DBConnector\data-export`: • `db2.properties` • `oracle.properties` In `DBConnector\data-import`: • `db2.properties` • `oracle.properties` In `DBConnector\sql-syntax-mapping`: • `DBSyntax.table.mssql.properties` • `DBSyntax.table.oracle.properties`

Configuring integration toolkit security

Shared secret-based authentication

You can get the shared secret configured by your Administrator from the Integration Toolkit Security page in Ariba Administrator.

Certificate-based authentication

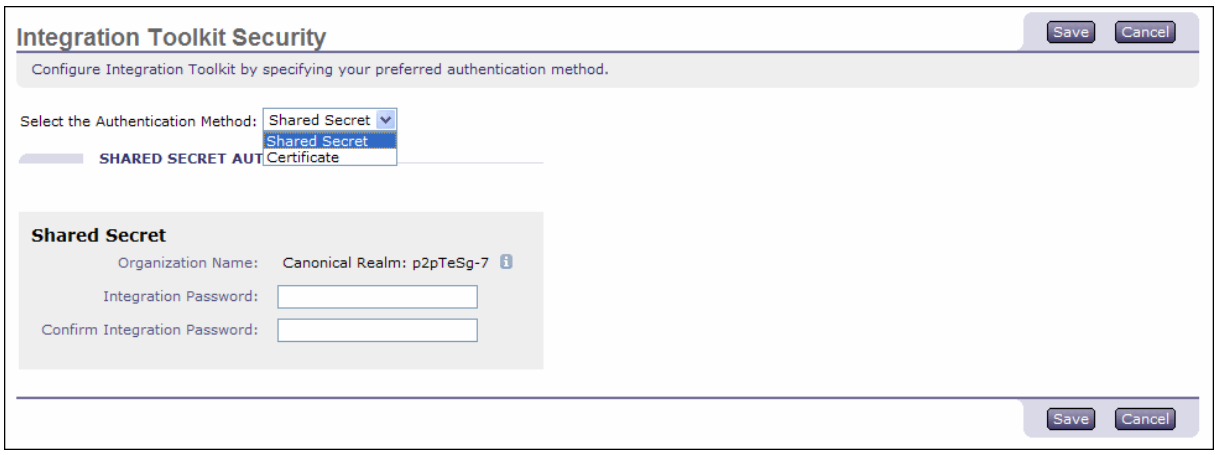
To ensure that the data exchange between SAP Ariba cloud solutions and SAP Ariba integration toolkit is completely secure, SAP Ariba provides digital certificate-based authentication. In SAP Ariba cloud solutions that are configured for certificate-based authentication, the SAP Ariba server uses the certificate to authenticate upload and download requests from the SAP Ariba integration toolkit.

> **Note**
>
> You must be a member of the Integration Administrator group to access this page in Ariba Administrator.

Using shared secret-based authentication

Procedure

1. In Ariba Administrator, under **Integration Manager** workspace, click **Integration Toolkit Security**. workspace.



2. Under **Select the Authentication Method**, select **Shared Secret** from the pull-down menu.
3. Under **Shared Secret**, enter the shared secret twice to confirm it.
4. Click **Save**.

Using certificate-based authentication

Prerequisites

You must have the following items to use digital certificate-based authentication:

- A public/private key pair, stored in a Java KeyStore (JKS).
- An X.509 certificate with the public key from your public/private key pair. The X.509 certificate must be base64-encoded in Privacy Enhanced Mail (PEM) format and signed by a Certificate Authority (CA) trusted by SAP Ariba. To get information about the CAs currently trusted by SAP Ariba or to register a CA to be trusted by SAP Ariba, contact SAP Ariba Customer Support.

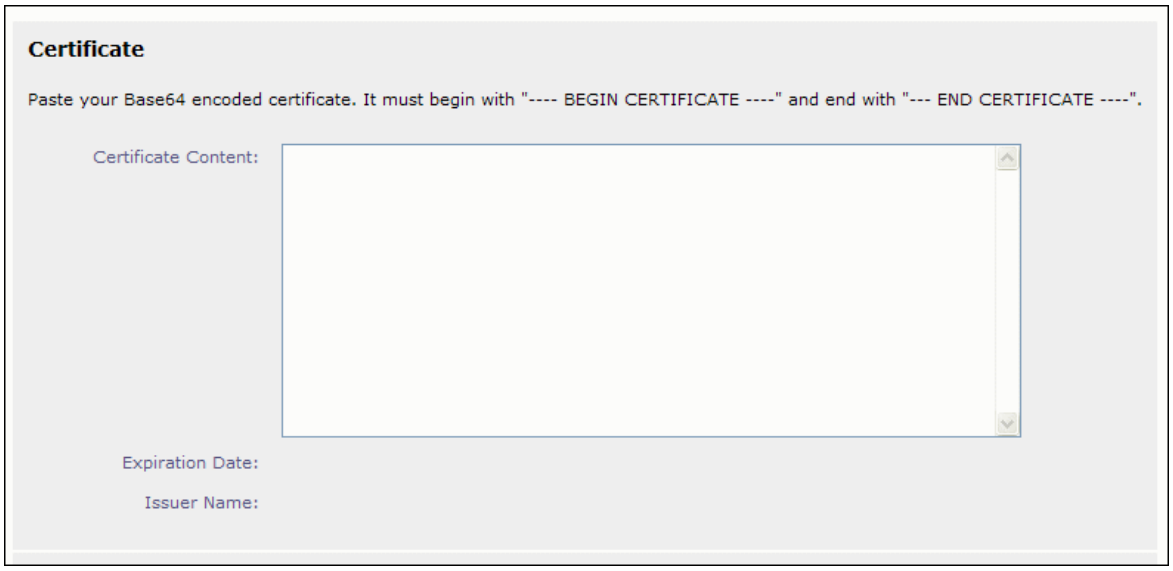
> **Note**
>
> - This procedure is not applicable when using the SAP Ariba integration toolkit on SAP NetWeaver. When using the SAP Ariba integration toolkit on SAP NetWeaver, you store encryption keys and security

certificates in the SAP NetWeaver PI Keystore, as described in Creating a View and Importing Certificates into SAP NetWeaver Keystore [page 104].

- The digital certificate must be from a certificate authority trusted by SAP Ariba. If you are using a certificate not recognized by SAP Ariba, then the SAP Ariba cloud solution does not accept requests from the SAP Ariba integration toolkit and an appropriate error is displayed.
- If you configure a new certificate when a certificate is already configured, the previous certificate is retained as a backup certificate. The SAP Ariba cloud solution uses the backup certificate to authenticate messages if authentication with the primary certificate fails. If the backup certificate has expired, then it is not displayed on the user interface.
- If the authentication is configured to use a digital certificate and the shared secret is passed as the authentication token from the SAP Ariba integration toolkit, an error is displayed.

Procedure

1. Use the `keytool -genkey` command to generate a public/private key pair (specify `-keyalg RSA`) and create a KeyStore. Make a note of the following items, which must be specified in the SAP Ariba data transfer tool options file:
    - the file path to the KeyStore; you will specify this in the SAP Ariba data transfer tool`clientKeyStore` option
    - the password for the KeyStore; you will specify this, encrypted, in the `clientKeyStorePassword` SAP Ariba data transfer tool option
    - the key password; you will specify this, encrypted, in the `clientKeyPassword` SAP Ariba data transfer tool option
2. Use the `keytool -certreq` command to generate a Certificate Signing Request (CSR) that references the KeyStore that contains your public/private key pair.
3. Submit the CSR to your Certificate Authority (CA) and request a X.509 certificate, base64-encoded in Privacy Enhanced Mail (PEM) format. (The first line of the certificate file is `-----BEGIN CERTIFICATE-----` and the last line is `-----END CERTIFICATE-----`).
4. Add your certificate to the SAP Ariba configuration.
    a. In Ariba Administrator, under **Integration Manager** workspace, click **Integration Toolkit Security**.
    b. Under **Select the Authentication Method**, select **Certificate** from the pull-down menu.
    c. In the **Certificate Content** field, paste the contents of your certificate, including the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` lines.



d. Click **Save**.

Using the -version and -info options

Use the `-version` and `-info` command line options to display version and environment information about the SAP Ariba integration toolkit.

- `-version` displays the version (for example, the content of the variables `ClientType` and `ClientVersion`.
- `-info` displays environment information (for example, the content of the variable `ClientTool`).

These options are not included in the wrapper script. The direct java - jar ... must be called to use these options.

For example, using the `-version` option as follows:

```
$ java -jar classes/aribafiletransfer.jar -version
```

results in the following output:

```
aribaintegrationtoolkit version : 10s2
```

Using the `-info` option as follows:

```
$ java -jar classes/aribafiletransfer.jar -info
```

results in the following output:

```
aribaintegrationtoolkit runtime environment :
network.identity : 10.10.62.79/jdoe
os.name : Windows XP
os.version : 5.1
os.arch : x86
java.vm.version : 1.5.0_12-b04
java.vm.vendor : Sun Microsystems Inc.
```

Related Information

Version Information Transfer [page 68]

SAP Ariba integration toolkit enhancements for Dell Boomi integration

SAP Ariba integration toolkit allows you to synchronize data between your ERP system and SAP Ariba cloud solutions, such as SAP Ariba Buying and Invoicing and SAP Ariba Sourcing. With this release, Dell Boomi (Boomi) can be used to run the SAP Ariba integration toolkit for synchronizing data between your ERP system and SAP Ariba cloud solutions. You can run both data upload and data download events from Boomi either manually or in a scheduled manner. The SAP Ariba integration toolkit captures the status of event runs in a CSV file that contains information about the success or failure of the executed events. To ensure that you can monitor SAP Ariba integration toolkit runs from Boomi, you can upload the generated CSV file to Boomi Monitoring. The generated CSV file contains the following columns:

- Status Code - contains the error or success code
- Status String - contains the error or success string
- Message - contains the detailed message for both error and success scenarios

The SAP Ariba integration toolkit contains the SAP Ariba data transfer tool and the DB Connector. However, SAP Ariba integration toolkit enhancements for Dell Boomi are used with the SAP Ariba data transfer tool only.

Enabling the integration of the SAP Ariba integration toolkit with Boomi

To enable the integration of the SAP Ariba integration toolkit with Boomi, you must:

- Upgrade to the latest version of the SAP Ariba integration toolkit. See Installing or Upgrading the Integration Toolkit [page 10].
- In Boomi, perform the required administration and configuration activities to ensure you can run the SAP Ariba integration toolkit and read the generated CSV file for monitoring purposes.

SAP Ariba integration toolkit from Boomi requirements

A new optional parameter called `monitorLogDirectory` is introduced as part of the `additionalClientArgs` parameter in the sample options file. The value for this parameter is a location on your file system where the generated CSV files are placed after an SAP Ariba integration toolkit run. For example, `D:\Ariba\IntegrationToolkit\DataDownlaod\CSVFiles`.

> **Note**
>
> After running the event from Boomi, read the status file (CSV), and delete the CSV file from the configured location.

Running the SAP Ariba integration toolkit from Boomi

When you run the SAP Ariba integration toolkit from Boomi, you may encounter any of the following failure scenarios:

Failure on the client system

The failures on the client system are attributed to all failures that occur locally on the client during initialization of the event run, parsing of the parameters, and so on.

The error code that is logged in the CSV file for failures on the client system is `1`.

Unable to connect to the SAP Ariba system

This error occurs when there are connection related issues between the SAP Ariba server and SAP Ariba integration toolkit. Some of the possible causes for this error are as follows:

- Internal Server Error
- Bad Gateway
- Service Unavailable
- Gateway Timeout

The error code that is logged in the CSV file for this error is `2`.

Request rejected by the SAP Ariba system

This error occurs when the SAP Ariba system rejects the request sent by SAP Ariba integration toolkit. Some of the possible causes for this error are as follows:

- Authentication failure
- Invalid filename
- Invalid event name

The error code that is logged in the CSV file for this error is `3`.

> **Note**
>
> The status code that is logged in the CSV file for successful data transfer is `0`.

Troubleshooting the integration of the SAP Ariba integration toolkit with Boomi

Some of the common errors that you may encounter for all failure types are as follows:

Failure on the client system

If you have not specified values for some parameters, such as `set site=` or `set urlPrefix=`, then you might see the following errors the generated CSV file.

```
"STATUS_CODE","STATUS_STRING","MESSAGE"
"1","Failure on the Client System","error initializing properties:
java.io.IOException: illegal arguments: no value for the property -urlPrefix"
"1","Failure on the Client System","error initializing properties:
java.io.IOException: illegal arguments: no value for the property -site"
```

To resolve the above errors, specify values for all the relevant parameters and run the event again.

Unable to connect to the SAP Ariba system

The following error is reported when the SAP Ariba server is busy:

```
"STATUS_CODE","STATUS_STRING","MESSAGE"
"2","Unable to Connect to Ariba System", “Service Unavailable Throwable caught:
Unavailable to service Event ReceiptDataLoad: server busy”
```

Request rejected by the SAP Ariba system

When the request is rejected by the SAP Ariba system, some of the errors that may get logged in the generated CSV file are as follows:

- If you have not specified any value for the `set integrationPasswordKeyPath` parameter, the following error is reported:

  ```
  "STATUS_CODE","STATUS_STRING","MESSAGE"
  "3","Request Rejected by the Ariba System","Unauthorized:Throwable caught:
  Authentication Failure- Invalid Integration Password or Authentication mode is
  of type Certificate || No stack available."
  ```

- If you have not specified any value for the `set additionalServerArgs` parameter, the following error is reported:

```
"STATUS_CODE","STATUS_STRING","MESSAGE"
"3","Request Rejected by the Ariba System","Bad Request:Throwable caught: null
is not a valid event name. || No stack available."
```

To resolve the above errors, specify the required value for the parameters and run the event again.

- If you have specified an incorrect file name for one of the files (CompanyCode1.csv instead of CompanyCode.csv) part of a batch import event, the following error is reported:

```
"STATUS_CODE","STATUS_STRING","MESSAGE"
"3","Request Rejected by the Ariba System","Bad Request:Throwable caught:
Illegal filename: CompanyCode1.csv || No stack available."
```

To resolve the above error, change the filename from `CompanyCode1.csv` to `CompanyCode.csv` and run the event again.

Data transfer successful

If the data transfer is successful, then the generated CSV file appears as follows:

```
"STATUS_CODE","STATUS_STRING","MESSAGE"
"0","Success","Data Transfer Successful"
```

Using the SAP Ariba data transfer tool

In this section:

Understanding the SAP Ariba data transfer tool

The SAP Ariba data transfer tool is a command-line utility that allows data transfer in batch mode between a back-end system and your SAP Ariba system. Data is transferred in the form of comma-separated-value (CSV) files. This chapter describes how to set up and use the SAP Ariba data transfer tool to transfer data.

For example, you can use SAP Ariba data transfer tool to export expense reports from your SAP Ariba cloud solution to your back-end system to synchronize your data. You can also use the tool with a scheduler to load master data periodically from your back-end system to the SAP Ariba cloud solution. You can import master data only as a batch import task.

You can use the SAP Ariba data transfer tool to perform these tasks:

- **Upload master data to your SAP Ariba system:** Synchronize the SAP Ariba application with your back-end system by transferring a snapshot of the master data from your back-end system.
- **Upload transactional data to your SAP Ariba system:** Synchronize the SAP Ariba application with your back-end system by transferring transactional data incrementally from your back-end system.
- **Upload spend visibility and sourcing data to your SAP Ariba system:** Synchronize SAP Ariba Spend Analysis with your back-end system for reporting and analysis tasks by transferring analytical data files in ZIP format from your back-end system. You can also initiate data loads after the transfer is complete.
- **Download spend visibility data from your SAP Ariba system:** Synchronize your back-end system with SAP Ariba Spend Analysis by transferring star schema data by source system. By default, the SAP Ariba data transfer tool will download the latest star schema export stored on the server. However, you can also use it to invoke and download a new export.
- **Download transactional data from your SAP Ariba system:** Synchronize your back-end system with SAP Ariba by transferring transactional data from an SAP Ariba application. Transactional data that you can download includes expense reports, payment requests, and accruals.
- **Download supplier and supplier location data from your SAP Ariba system.** This also includes information about the suppliers bank.

The SAP Ariba data transfer tool is a HTTP client that resides on a system behind your firewall and runs on Microsoft Windows or UNIX. Wrapper scripts in the SAP Ariba data transfer tool use options that you specify in options files to run the required task.

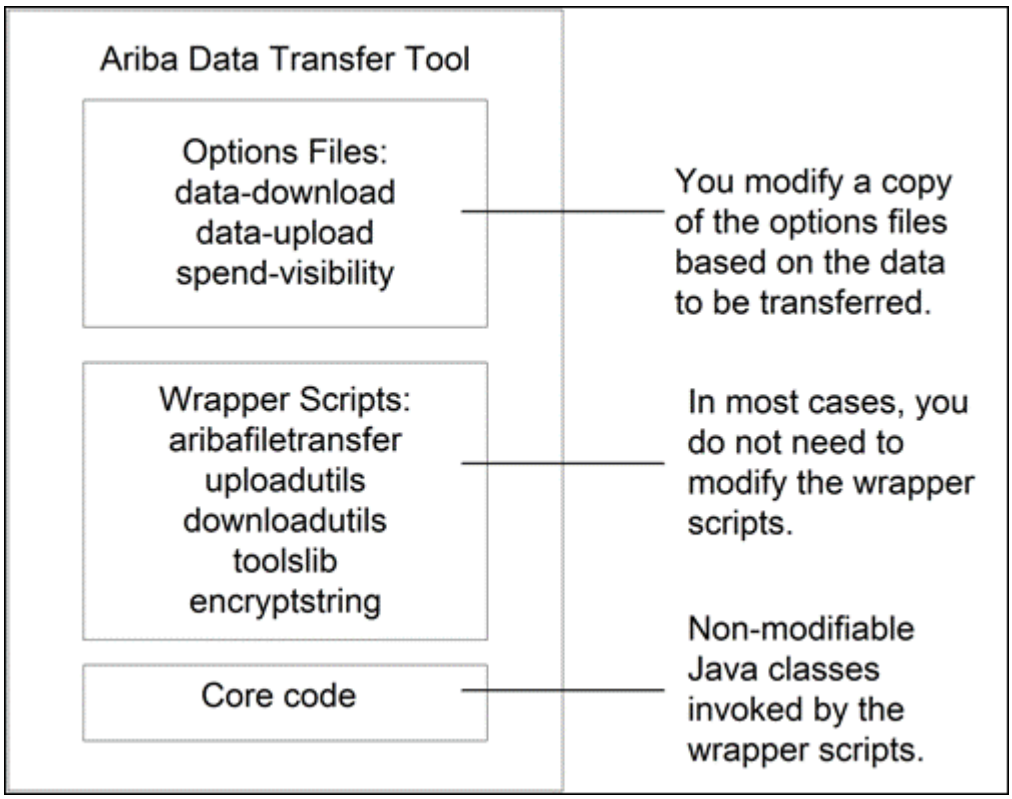


Figure 2: Ariba Data Transfer Tool Components

SAP Ariba data transfer tool Process Overview

To use the SAP Ariba data transfer tool, you set options in working copies of the sample options files. In the options files, you specify your company's site name, partition, authentication information, the location of data to be loaded, and a location to receive the data exported from the SAP Ariba application, along with other options.

When you start the SAP Ariba data transfer tool from the command line, the tool checks for the presence of required values and transfers the data to your site using an HTTPS post. The tool determines whether to use username and password, shared secret, or certificate and sends the correct credential to the service for authentication.

The options files also specify whether to send an email notification reporting the status of the data transfer and also whether to perform a cleanup of old data.

The options files are available for Microsoft Windows (as `.bat` files) and UNIX (as `.sh` files) and include:

- Data download sample files—for transferring transactional data to your site
- Data upload sample files—for transferring master data, incremental master data, and transactional status data to Ariba
- SAP Ariba Spend Analysis sample files—for transferring SAP Ariba Spend Analysis data (in ZIP files) to SAP Ariba and star schema data from Spend Visibility to your site.

The wrapper scripts provide the tool with the environment variables specified in the options files. The wrapper scripts are available for Microsoft Windows (as `.bat` files) and UNIX (as `.sh` files), and include:

- `aribafiletransfer`—the main script that performs the data transfer
- `downloadutils`—contains functions called by `aribafiletransfer` for transferring data from the Ariba application to your site
- `uploadutils`—contains functions called by `aribafiletransfer` for transferring data to the Ariba application from your site
- `toolslib`—contains helper utilities for the other wrapper scripts
- `encryptstring`—an optional script that encrypts the shared secret and, for Spend Visibility, the `password` option.
- `encryptcustomerkey` - a script that allows you to encrypt a customer key that is then used by `encryptstring`.

Related Information

Customizing the Sample Options Files [page 26]
About Security Encryption [page 24]

CSV file descriptions

The most comprehensive source of information about the content and format of the CSV files used to import and export data from your SAP Ariba system is the data dictionary, which contains precise descriptions of the columns in each CSV file on individual worksheets by task. For instructions on how to download and navigate the data dictionary, see the Data Load Guide for your SAP Ariba solution.

For more information about CSV files, as well as a general primer on CSV file formatting, see the *data import and administration guide* for your SAP Ariba solution.

Configuring the SAP Ariba data transfer tool

This section describes the steps that you must perform to set up and run the SAP Ariba data transfer tool, including:

- Encrypting your shared secret or password for security purposes
- Customizing copies of the sample options files
- Running the SAP Ariba data transfer tool from the command line, or setting up the scheduler of your choice to run the tool. For more information, refer to the documentation for your scheduler.

> **Note**
>
> SAP Ariba data transfer tool supports a minimum read timeout of 5 minutes for data downloads from the SAP Ariba server. If for example, the read timeout is set to 3 minutes and the data download takes more than 3

minutes, then the connection is timed out at the client side. Therefore, to prevent a read timeout, ensure that the read timeout is set to a minimum of 5 minutes.

> **Note**
>
> Running data transfer events without sufficient time interval causes errors. SAP Ariba recommends that you schedule a minimum interval of 30 minutes between subsequent data transfer events.

About security encryption

For enhanced security, you can encrypt the following parameters used in the options file:

- `integrationpasswordKeyPath` (a file that contains an encrypted copy of the shared secret configured on the **Integration Toolkit Security** page in Ariba Administrator)
- `clientKeystorePassword`
- `clientKeyPassword`
- `proxyPassword`

The SAP Ariba data transfer tool includes two utility scripts you use together to create encrypted strings so that the key used to encrypt the strings is also encrypted:

- `encryptcustomerkey`
- `encryptstring`

You use the `encryptcustomerkey` script to create an encrypted key that you store in a local file. This script prompts you for a 24-character (192 bits) customer key that is encrypted using the Triple Data Encryption Algorithm (3DES) with three distinct 56 bit keys. The SAP Ariba data transfer tool uses a single key of required length that the 3DES breaks up appropriately into the required keys for encryption.

You use the `encryptstring` script to encrypt a string (such as the shared secret for the SAP Ariba integration toolkit) that you store in a local file. This script prompts you for the string and uses the key encrypted by `encryptcustomerkey` to encrypt the string.

All SAP Ariba integration toolkit scripts must be able to resolve the location of `java.exe`. You can use the `JAVA_HOME` option in your options file to specify the path to `java.exe`.

Encrypting SAP Ariba data transfer tool parameters

Procedure

1. Install Java Runtime Environment 1.6, if it is not already installed.
2. Find the directory containing `java.exe`. For example:

   ```
   C:\Program Files\Java\jre1.x\bin
   ```

3. Open the options file you intend to use in a text editor and find the `JAVA_HOME` option.

4. Set the `JAVA_HOME` option to the path up to, but not including the bin directory. For example:

   ```
   set JAVA_HOME=C:\Program Files\Java\<jre version>
   ```

5. At the command prompt change to the directory containing the `encryptcustomerkey` script. For example:

   ```
   cd D:\ITK\bin
   ```

6. Run the script to encrypt the customer key using the following syntax:

   ```
   encryptcustomerkey.bat optionsFile
   ```

   where `optionsFile` is the path and name of an options file where you have set the `JAVA_HOME` option.

   For example: `D:\ITK\bin\encryptcustomerkey.bat ..\myConfig\POtransactionalcust.bat`

   A prompt to enter the key is displayed.

7. Enter the customer encryption key to be encrypted.

   The key must be an alphanumeric string of 24 characters in length. For example, `TestCustomerKey123456789`.

   The encrypted key is displayed as an output on the command line. For example, `{3DES}+Y9xa4mwCKKvBh6oPRIFvkdk2srHINb/I0JygsfYQWQ=`.

8. Copy the encrypted key from the command line to a text file and save the file. For example, `Password.txt`.

   The encrypted key must be the first and only line in the file without any spaces before or after. The path to this file must be passed as the value for the `encryptionKeyPath` parameter in your options file.

9. In the transactional batch file for example, `POtransactionalcust.bat`, pass the path of the file containing the key (for example, `Password.txt`) for the `encryptionKeypath` parameter in the options file. For example:

   ```
   set encryptionKeyPath=D:\ITK\Password.txt
   ```

10. Save transactional batch file.
11. Run the script to encrypt a string using the following syntax:

    ```
    encryptstring.bat optionsFile
    ```

    where `optionsFile` is the path and name of an options file where you have set the `JAVA_HOME` option.

    For example: `D:\ITK\bin\encryptstring.bat ..\myConfig\POtransactionalcust.bat`

    A prompt to enter the string is displayed.

12. Enter the string you want to encrypt, such as the shared secret configured on the **Integration Toolkit Security** page in Ariba Administrator.

    The encrypted string is displayed as an output on the command line. For example, `{3DES}xXk8vlgZl6iVmdCJYy/n1w=`

13. Copy the encrypted string from the command line to a text file and save the file. For example, `encryptedSharedSecret.txt`.

    The encrypted string must be the first and only line in the file without any spaces before or after.

14. In the transactional batch file, for example, `POtransactionalcust.bat`, set the path of the file containing the encrypted string (`encryptedSharedSecret.txt` in this example) in the appropriate parameter (the `integrationPasswordKeypath` parameter in this example):

    ```
    set integrationpasswordKeyPath=D:\ITK\encryptedSharedecret.txt
    ```

15. Save transactional batch file.

Next Steps

You can now run the command to download data:

```
D:\ITK\bin\aribafiletransfer.bat ..\myConfig\POtransactionalcust.bat
```

Customizing the sample options files

Context

Before using the SAP Ariba data transfer tool, you must set options in working copies of the sample options files. The sample options file that you use is based on the task you want to perform.

Procedure

1. Navigate to the `sample-options/DataTransferTool` directory that was created when you extracted the SAP Ariba integration toolkit jar file. The sample options files are grouped by task:

Directory	Options Files
`data-download` (export)	○ `transactionaldata.bat.sample` ○ `transactionaldata.sh.sample` ○ `wsdata.bat.sample` (to export contract workspaces from a Windows client) ○ wsdata.sh.sample (to export contract workspaces from a Unix client)
`data-upload` (import)	○ `incremental-masterdata.bat.sample` ○ `incremental-masterdata.sh.sample` ○ `masterdata.bat.sample` ○ `masterdata.sh.sample` ○ `transactionaldata.bat.sample` ○ `transactionaldata.sh.sample`
`spend-visibility` (import)	○ `upload.bat.sample` ○ `upload.sh.sample`

Directory	Options Files
`sv-download` (export)	○ `download.bat.sample` ○ `download.sh.sample`

> **Note**
>
> To customize the SAP Ariba Spend Analysis export sample options files, navigate to the `sample-options/DataTransferTool/sv-download` directory.

2. Make a copy of the appropriate sample options file, move the copy to another directory, and rename the file.

> **Note**
>
> If you are running the tool on Windows, the options files must end in `.bat`. For example, the Microsoft Windows file for SAP Ariba Spend Analysis (`upload.bat.sample`) must be renamed `upload.bat` or the wrapper script will not run.

3. Use a text editor to modify the sample files you just copied. See Ariba Data Transfer Tool Options [page 27].

> **Note**
>
> If you are running the tool on Microsoft Windows, double quotes are required for two options, `event` and `sourceSystem`.

Related Information

Installing the Ariba Integration Toolkit [page 11]

SAP Ariba data transfer tool options

This section describes each option that appears in the sample options files, and the operation for which it is used. Note that the options appear in slightly different order based on the sample file you are using.

> **Caution**
>
> In each sample options file, there is a section marked “Do not modify anything after this line.” For SAP Ariba cloud solutions, this section of the file is pre-configured, and must not be changed except when you are using SAP Ariba Sourcing, SAP SAP Ariba Contracts, Ariba Supplier Information and Performance Management, and SAP Ariba Spend Analysis solutions. In such cases, you must set the `urlPrefix` to the appropriate value for your system.

The following table describes the options you specify in the sample options files:.

Option	Description	Used For
JAVA_HOME	Specify the full path to the Java JRE executable, `java.exe`, up to, but not including the `bin` directory. Example: `set JAVA_HOME`=C:\Program Files (x86)\Java\jre6 or C:\Program Files\Java\jre7 This environment variable is used by the `encryptstring.bat (sh)` script and other wrapper scripts, which use Java.	Upload, Download
site	Specify the name of your company as used in the URL for your SAP Ariba product. The URL is in the for `http://mycompany.procurement.ariba.com.` Example: `set site=mycompany`	Upload, Download, Spend Visibility
serviceMode	Specifies what kind of SAP Ariba product you are using: Possible values include: • `0` to specify an SAP Ariba cloud solution. The `urlPrefix` option will, in this case, refer to a site. For example: `http://s1.ariba.com/Buyer/fileupload?realm=realm_1` • `-1` is the default value, which must be changed to the appropriate allowed value to avoid a warning message. The warning message will be issued to the client log, followed by a permanent failure 4xx error code.	Upload, Download, Spend Visibility
authUser	Use to specify the user name (required for SAP Ariba Spend Analysis uploads). Do not specify `authUser` for master data uploads or transactional data uploads and downloads.	SAP Ariba Spend Analysis
password	Use to specify your password (required for SAP Ariba Spend Analysis uploads). Your customer administrator can provide a password. Do not specify `password` for master data uploads or transactional data uploads and downloads. SAP Ariba recommends that you encrypt the value for this option using the `encryptcustomerkey` and `encryptstring` utilities.	SAP Ariba Spend Analysis
integrationPasswordKeyPath	This specifies the location of the file that contains the shared secret configured on the **Integration Toolkit Security** page in Ariba Administrator. SAP Ariba recommends that you encrypt the contents of this file using the `encryptcustomerkey` and `encryptstring` utilities.	Upload, Download
encryptionKeyPath	`encryptcustomerkey`. The contents of this file are used by the This specifies the location of the file that contains the encrypted customer key created by `encryptstring` utility.	Upload, Download
clientKeystore	Used for certificate-based authentication. The path to the Java KeyStore that contains the public/private key pair associated with your client certificate.	Upload, Download, Spend Visibility

Option	Description	Used For
`This specifies the location of the file that containsclientKeyStore Password`	Used for certificate-based authentication. The password to access the Java KeyStore specified by the clientKeystore option. SAP Ariba recommends that you encrypt the value for this option using the `encryptcustomerkey` and `encryptstring` utilities.	Upload, Download, Spend Visibility
`clientKeyPassword`	Used for certificate-based authentication. The password to access the public/private key pair in the Java KeyStore specified by the client-Keystore option. SAP Ariba recommends that you encrypt the value for this option using the `encryptcustomerkey` and `encryptstring` utilities.	Upload, Download, Spend Visibility
`event`	Use to specify an event. To use an event with the SAP Ariba data transfer tool, the event requires configuration on the server side. You must specify the SAP Ariba integration toolkit name of an event and not the display name. The SAP Ariba integration toolkit name is available in the Data Import / Export page in Ariba Administrator. If you are using Microsoft Windows, you must quote the value of the `event` option. For more information about data import and export tasks, see the implementation guide for your SAP Ariba solution. For information about the form and contents of CSV files, use the Data Dictionary link in Ariba Administrator to download the data dictionary.	Upload (transactional only), Download
`adaptersource`	The value of adapter source can be ALL, EXTERNAL, SYSTEM, or ARIBA MANAGED.	Download
`dataFile`	Use to specify a directory of your choice. This directory can only contain either a single ZIP file or CSV files but not both together. You can also specify the path directly to a ZIP file or a CSV file.	SAP Ariba Spend Analysis
`downloadDir`	Use to specify the location where the star schema export file is stored (required for SAP Ariba Spend Analysis downloads).	SAP Ariba Spend Analysis
`lastModified`	Use to specify that the export will only include data modified after the specified date in the format DDMMYYYY (required for SAP Ariba Spend Analysis downloads). If you do not use this option, the export includes all data.	SAP Ariba Spend Analysis
`factTables`	Use to specify the fact tables to include in the export in a comma-separated list (required for SAP Ariba Spend Analysis downloads). If you do not use this option, the export includes all fact tables.	SAP Ariba Spend Analysis
`sourceSystem`	Use to specify the source system in a site that the file is for (required for SAP Ariba Spend Analysis uploads) or to specify the source systems to include in the export in a comma-separated list (required for SAP Ariba Spend Analysis downloads). If you do not use this option, the export includes all source systems. Quotes are required for the value of this option in Microsoft Windows.	SAP Ariba Spend Analysis

Option	Description	Used For
`generateSchemaExport`	Use to specify whether or not to start a new star schema export (required for SAP Ariba Spend Analysis downloads). By default, `generateSchemaExport` is set to `false`, and the SAP Ariba data transfer tool downloads existing export files from the specified location. If you set this option to `true`, the SAP Ariba data transfer tool starts a new star schema export operation using the `lastModified`, `factTables`, and `sourceSystem` options you specify and then downloads the results.	SAP Ariba Spend Analysis
`downloaddir`	Use to specify the directory (named `DownloadFromAriba` by default) containing the downloaded files in directories to be stored for processing by your back-end system. On Microsoft Windows, the directory is created in `C:\TEMP`. On UNIX, it is created in `/var/tmp`.	Download
`additionalClientArgs`	Specify the following variables in the parameter `additionalClientArgs` to configure the upload and download behavior between the SAP Ariba Buying solutions and your back-end system: • `unzipDirs`: Use to specify if the downloaded ZIP file is to be unzipped. • `downloadInZipFormat`: Use to specify if you want to download responses sent from the SAP Ariba Buying solutions in a zipped format. The default value is `false`. • `numberOfReConnectionsToServer`: Use to specify the number of retries. • `waitTimeBeforeReconnection`: Use to specify the wait time between retries in milliseconds. For example, `set additionalClientArgs=-unzipDirs false -downloadInZipFormat false -numberOfReConnectionsToServer 5 -waitTimeBeforeReconnection 30000`	Upload, Download
`waitTimeBetweenIncUpload`	Use to specify wait time in minutes between incremental uploads.	Upload
`timestampFile`	Use to specify, for incremental downloads, the time stamp file that ensures the download event pulls data saved since the latest successful download.	Download
`filePrefix`	Use to specify a prefix for downloaded folders, for example `expense`, `payment`, or `accruals`. The tool appends a time stamp to downloaded files, indicating when the download was completed. For example, you can specify the following prefixes based on the transaction: • ERPOrder for PO • ERPOrderChange for change PO • ERPOrderCancel for cancel PO • Receipt for receipt • Payment for payment • ExpenseReport for expense	Download

Option	Description	Used For
`notifyemail`	Specify a comma-delimited list of email addresses to which notifications must be sent, if you have enabled notifications for either successful or failed data transfers. Example: `set notifyemail=myemail@mycompany.com`	Upload, Download, Spend Visibility
`smtpMailServer`	Specify the Simple Mail Transfer Protocol (SMTP) server that the SAP Ariba data transfer tool email client will use to send notification messages. Example: `set smtpMailServer=mail.mycompany.com`	Upload, Download, Spend Visibility
`fromEmail`	The from address of the notification email. It is mandatory to specify a valid from address.	Upload, Download
`topDir`	Specify the full path to the directory where the various input and output subdirectories must go. Example: `set topDir=C:\TEMP\FullMasterUpload`	Upload
`inDir`	The folder in which data files will be placed for upload to the SAP Ariba cloud solution. Specify the full path to the subdirectory containing the CSV files to be sent. The tool expects to find a subdirectory containing the CSV files to be transferred. Example: `set inDir=%topDir%\InDir \your_subdirectory` For example, in an Ariba Buyer instance with two partitions, you can specify the subdirectory this way: `set inDir=%topDir%\InDir\partition2`	Upload
`filterprefix`	Specify the subdirectory name prefix that indicates which CSV file subdirectories (in `inDir`) the tool must ignore when uploading files. Default: `set filterprefix=processing` If you have a program that generates CSV files, it must use a file name that begins with this prefix and then rename the file when the file is ready for transfer. If this task is scheduled to run while the file is still being generated, the file is ignored until the next scheduled run.	Upload
`outDir`	Specify the full path to the directory where the SAP Ariba data transfer tool moves CSV files after it sends copies to the SAP Ariba cloud solution. Default: `set outDir=%topDir%\OutDir`	Upload

Option	Description	Used For
`unsentDir`	When you have an automatic mechanism to create full master CSV files, and it creates a new batch before your SAP Ariba data transfer tool has sent the old batch, the old batch does not need to be sent. Specify a directory in which to put unsent data. Default: `set unsentDir=%topDir%\UnsentDir` This option is only present in the `masterdata` options file. With incremental data, newer and older files may contain different data, so all files are sent and this option is absent.	Upload (master data only)
`PerformCleanup`	Specify `1` if you want the SAP Ariba data transfer tool to delete sent CSV files from the `outDir` directory when they are older than the `cleanupFilesOlderThan` value. Default: `set PerformCleanup=1`	Upload
`cleanupFilesOlderThan`	Specify the maximum number of days you wish to retain sent CSV files in the `outDir` directory. Default: `set cleanupFilesOlderThan=7`	Upload
`logDir`	Specify the full path to the directory where the SAP Ariba data transfer tool writes a log file after it finishes the task that uses this options file. Default: `set logDir=%topDir%\UnsentDir`	Upload, Download, Spend Visibility
`notifyonsuccess`	Specify 1 to enable sending an email message when the data transfer is successful. Specify 0 to disable this feature. Default `set notifyonsuccess=1`	Upload, Download, Spend Visibility
`notifyonfailure`	Specify 1 to enable sending an email message when the data transfer is not successful. Specify 0 to disable this feature. Default `set notifyonfailure=1`	Upload, Download, Spend Visibility
`notificationSuccessSubject`	Specify the subject of the notification email message when the data transfer is successful. For example: `set notificationSuccessSubject=AribaFileTransfer Master Data Upload: Success`	Upload, Download, Spend Visibility
`notificationFailureSubject`	Specify the subject of the notification email message when the data transfer fails. For example: `set notificationFailureSubject=AribaFileTransfer Master Data Upload: Failed`	Upload, Download, Spend Visibility

Option	Description	Used For
`returnExitStatus`	Specify to support the return codes for Windows users. Windows users that need to perform some action on the exit status in the SAP Ariba integration toolkit must add the parameter to the sample-options file as follows: set `returnExitStatus=1` When the parameter is set to 1, the SAP Ariba integration toolkit returns the exit status to the caller. This is an optional parameter.	Upload, Download Spend Visibility

Related Information

About security encryption [page 24]
Certificate-based authentication [page 13]
Configuring events for use with the SAP Ariba data transfer tool [page 46]

Configuring proxy server options

The SAP Ariba data transfer tool can be configured to use Secure HyperText Transfer Protocol (HTTPS) through a proxy server. The following methods of authentication on the proxy server are supported:

- Basic
- Digest
- NTLMv1 (Note that NTLMv2 is not supported)

To enable secure communication through a proxy server, configure the following options:

Option	Description	Used For
`proxyHost`	Host name or IP address of the proxy server. Example: `set proxyHost=localhost`	Upload Download Spend Visibility
`proxyPort`	Port used by the proxy server to handle secure communication. Example: `set proxyPort=8081`	Upload Download Spend Visibility

Option	Description	Used For

`proxyAuthMod`	Indicates the method used for authentication on the proxy server. If the option is not defined, then no authentication is used with the proxy server.
	Note that NTLMv2 is not supported.
	Possible values:
	`B: basic`
	`D: digest`
	`N: NTLM`
	For example: `set proxyAuthMode=D`
		Upload
		Download
		Spend Visibility

`proxyUser`	The user name used for authentication. Required for all three authentication methods. For example: `set proxyUser=username`	Upload Download Spend Visibility
`proxyPassword`	The password used for authentication. Required for all three authentication methods. You can encrypt this password as you encrypt the shared secret. See About Security Encryption [page 24]. For example: `set proxyPassword=password`	Upload Download Spend Visibility
`proxyDomain`	The domain to which the user identified by the `proxyUser` option belongs. Required for NTLM authentication only. For example: `set proxyDomain=ARIBA`	Upload Download Spend Visibility

Configuring SAP Ariba Spend Analysis data load options

You can configure the SAP Ariba data transfer tool to load SAP Ariba Spend Analysis data after file upload, either by adding to the next pending data load or by creating a new data load.

For details on data loading operations and post-processing tasks, see the *Ariba Reporting and Analysis Data Load and Administration Guide*.

> **Note**
>
> Do not load data without consulting your SAP Ariba Spend Analysis Project Manager. Project Managers coordinate data load schedules and are aware of other data load activity, such as enrichment refreshment loads, that might conflict with automatic loads.

To load SAP Ariba Spend Analysis data after file upload, configure the following options:

Option	Description	Used For
`loadOperationName`	Specify the name of the new data load operation to which to add the uploaded data files. If you do not use this option, the uploaded data file is added to any load operation. For example: `set loadOperationName=ADTT`	SAP Ariba Spend Analysis
`startLoadOperation`	Set to `true` to schedule a data load after file upload is complete. If you do not use this option, the data load is not scheduled.	SAP Ariba Spend Analysis
`postProcess`	Set to `true` to run post-processing tasks after the data load. If you do not use this option, the data load does not include post-processing tasks. For example: `set postProcess=true`	SAP Ariba Spend Analysis
`indexDrop`	Set to true to drop and rebuild all indexes during the data load. If you do not use this option, the data load drops and rebuilds the indexes by default. For example: `set indexDrop=false`	SAP Ariba Spend Analysis
`applyExisting Enrichment`	Set to `true` to apply existing enrichment results to the data in the data load. If you do not use this option, existing enrichment results are not applied. For example: `set applyExistingEnrichment=true`	SAP Ariba Spend Analysis
`forceRecalcMV`	Set to `true` to recreate all materialized views whether or not underlying tables were modified during the data load. If you do not use this option, only those materialized views with underlying tables were modified during the data load are recreated. For example: set forceRecalcMV	SAP Ariba Spend Analysis
`additionalServer Args`	Set arguments for all of the data load options you want to use, including the source system. For example: `set additionalServerArgs=-sourceSystem %SourceSystem% -loadOperationName %loadOperationName% -startLoadOperation %startLoadOperation% -postProcess %postProcess% -indexDrop %indexDrop% -applyExistingEnrichment %applyExistingEnrichment% -forceRecalcMV %forceRecalcMV`	SAP Ariba Spend Analysis

Configuring contract workspace export options

The Export Workspace Data service can be used to download contract workspace data. Customers can download contract workspace data periodically or one time and save the data on their systems. Customers can also choose to download contract workspace data by the date they were last modified.

The output of the Export Workspace Data service is for archival purposes only. SAP Ariba reserves the right to change the contents of this output.

> **Note**
>
> To export contract workspaces in a ZIP format, the user must be an Integration Manager or have permissions to export contract workspaces.

Two options files are added to support the new export of the contract workspaces functionality and placed in the `data-download` folder in the SAP Ariba integration toolkit.

Exporting deleted contract workspaces

The Contract Workspace Export feature allows you to export deleted contract workspaces. When the `wsIncludeDeletedWorkspaces` parameter is enabled, deleted contract workspaces are included in the Contract Workspace Export.

The **Active** column in the Index.csv file displays **Yes** or **No**, depending on whether the contract workspace is active or has been deleted.

> **Note**
>
> The Contract Workspace Export feature allows you to export deleted contract workspaces, it does not enable you to export deleted contracts.

When deleted contract workspaces are exported, the SAP Ariba solution appends "_deleted" to the exported contract workspace ZIP file name. For example, if you export a deleted contract workspace named "CW2005", SAP Ariba solution exports a ZIP file named "CW2005_deleted.zip". None of the files within the ZIP file are appended with "_deleted", only the ZIP file name.

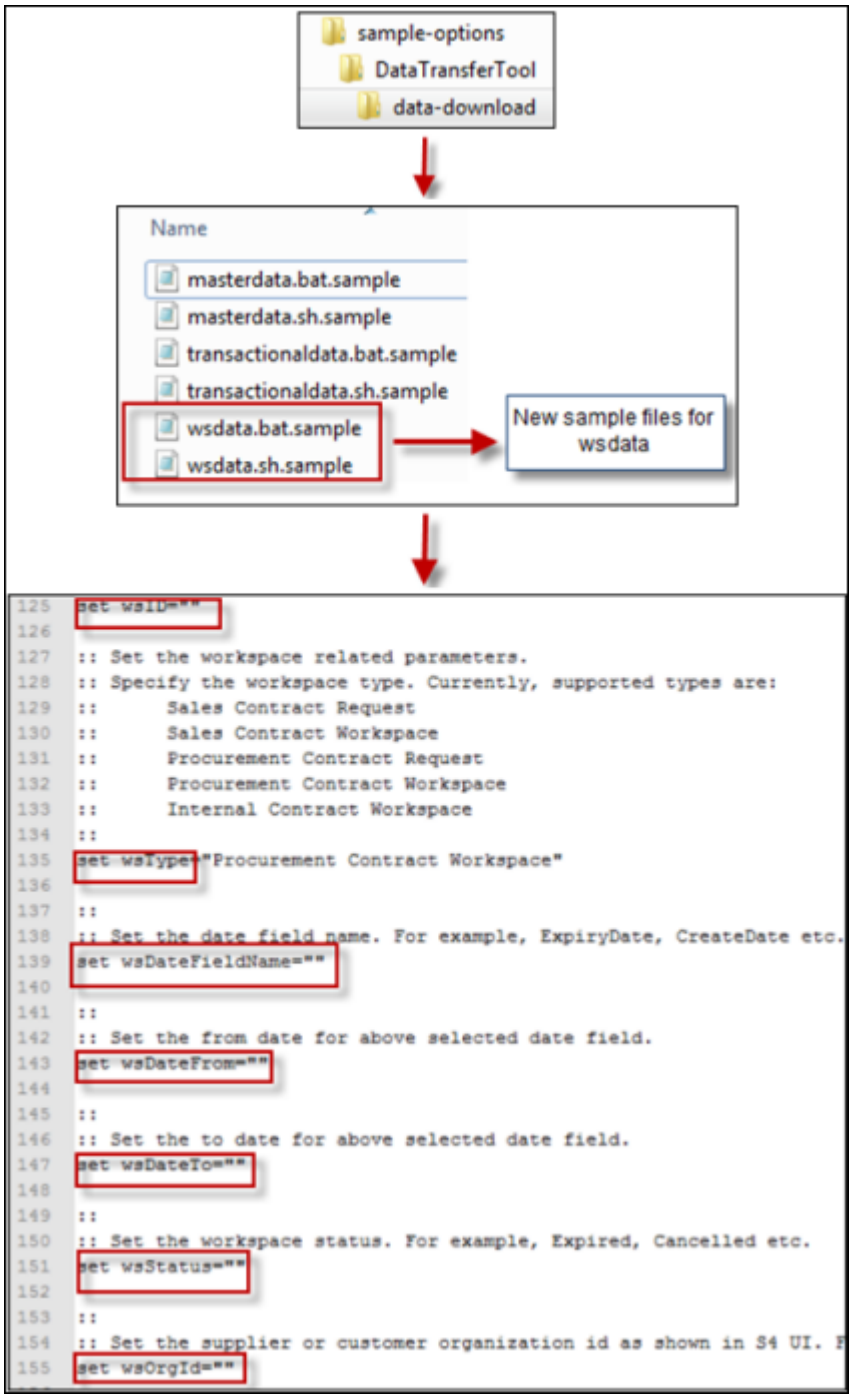
Configuring the options file

Procedure

1. Navigate to the `sample-options/DataTransferTool/data-download` directory. Two new options files are added:

Directory	Newly-Added Options Files
`//sample-options/DataTransferTool/data-download/`	`wsdata.bat.sample (Windows)` `wsdata.sh.sample (Unix)`

2. The newly-added options files in the `Data Transfer Tool` directory appear in the `data download` directory as follows:



3. Make a copy of the appropriate sample options file, move the copy to another directory, and rename the file.
    - For Windows clients, copy the `wsdata.bat.sample file.`
    - For Unix clients, copy the `wsdata.sh.sample` file.
4. Use a text editor to modify the sample file you just copied. Set the appropriate options as needed for downloading data.In addition, set options that are specific to contract workspace downloads as listed in the table below.

> **Note**
>
> - The table consists of a list of parameters in the options file and their description. For errors and recommended solutions, see Configuring events for use with the SAP Ariba data transfer tool [page 46]. If the problem persists, have your Designated Support Contact log a service request and an SAP Ariba Customer Support representative will follow up to complete the request.
> - You can use multiple combinations of options to export data. See Site configuration options for contract workspace exports [page 45].

5. Access the "index" CSV file for field values for each workspace and the location of the corresponding ZIP file.

Options file parameters

Parameter	Description
`event`	Specify `"Export Workspace Data"` to export contract workspace data.

`wsID`	Provide a valid contract ID as shown in the Overview tab in the contract workspace.
	Examples:
	`wsID = "CW1221"`
	`wsID = "CW1221, CW3715, CW2687"`
	Use a comma-separated list for multiple workspace IDs. Do not specify more than 1000 values.
	If you specify `wsID`, you cannot perform incremental exports (`wsExportType=Incremental` is ignored).

`wsType`	Select a workspace type from the following supported types to export the required data:
	• `Sales Contract Request`
	• `Sales Contract Workspace`
	• `Procurement Contract Request`
	• `Procurement Contract Workspace`
	• `Internal Contract Workspace`
	Examples:
	`wsType = "Procurement Contract Workspace"`
	`wsType = "Sales Contract Request, Sales Contract Workspace, Procurement Contract Request, Procurement Contract Workspace"`
	Use a comma-separated list for multiple workspace types.
	If you specify `wsType`, you cannot perform incremental exports (`wsExportType=Incremental` is ignored).

Parameter	Description

`wsDateFieldName`
 	
`wsDateFrom="YYYY/MM/DD"`
 	
`wsDateTo="YYYY/MM/DD”`
	Use the `wsDateFieldName, wsDateFrom`, and `wsDateTo` options together to define the date filters for the export.
	
	The wsDateFieldName option specifies the names of the date fields in the contract to be bounded by the date range (`wsDateFrom and wsDateTo` option values).
	
	Valid values for `wsDateFieldName`:
	
	• `Agreement Date`
	• `Create Date`
	• `Effective Date`
	• `Expiration Date`
	• `Notice Date`
	• Last Modified Date
	Use a comma-separated list for multiple `wsDateFieldName` values.
	
	Note
	If you specify multiple values for wsDateFieldName, the date range applies to all specified date fields. You can specify only one pair of values for the date range (`wsDateFrom and wsDateTo`). Each value for wsDateFieldName is combined with the wsDateFrom. and wsDateTo values to define a date filter. If there are multiple date filters, a contract workspace is exported only if it meets any of the date filter requirements.
	
	Example:
	
	`wsDateFieldName="Effective Date, Expiration Date"`
	
	`wsDateFrom="2012/01/31"`
	
	`wsDateTo="2013/12/31"`

`wsStatus`	Specify a status for a workspace from the following available values: • `Closed` • `Draft Amendment` • `Draft` • `Expired` • `On Hold` • `Pending` • `Published` • `Publishing` Example: `wsStatus=”Closed, Draft”` Use a comma-separated list for multiple statuses.

Parameter	Description
`wsOrgId`	Specifies the unique ID for the supplier or customer organization. You can access the ID by viewing the Organization Profile page for a supplier or customer. Example: `wsOrgId="ACM12345, ACM6789"` Use a comma-separated list for multiple IDs. Do not specify more than 1000 values. **Note** This option is not applicable to Internal Contract Workspaces.
`wsExportType`	Specifies the type of export. Valid values: - `All` (export all workspaces to your site) - `Incremental` (export only the contract workspaces created or modified after the last export) **Note** If `wsExportType` is `Incremental`, you must also specify the `timestampFile` option. If `wsExportType` is `Incremental` and the file specified for `timestampFile` is empty or does not exist (indicating that this is the initial export), all contract workspaces are exported. Subsequent exports will use the contents of `timestampFile` to export only the contract workspaces created or modified after the previous export. You cannot perform incremental exports if you specify `wsID` or `wsType` values (`wsExportType=Incremental` is ignored if you specify `wsID` or `wsType` values). Example: `wsExportType=All`
`timestampFile`	Specifies the location of the timestamp file for incremental downloads. See the `wsExportType` option for more information. Example: `timestampFile=%downloadDir%\WorkspaceTimeStamp`

Parameter	Description
`rerun`	If a workspace export operation is cancelled or interrupted before it is complete, you can set `rerun="true"` and resubmit your request. The SAP Ariba integration toolkit server will resume the previous export operation to the same output directory. Note that the server resumes the export operation after the last file successfully exported; the server does not resend files successfully exported in the previous operation. If this option is `true` and the previous workspace export operation was successful, the SAP Ariba integration toolkit displays an error message. If this option is `true` but any workspace filter options (`wsnnn` options, such as `wsStatus`) do not match the options for the incomplete export, the SAP Ariba integration toolkit displays an error message. The default value for this option is `false`. Example: `rerun="false"`
`retryLimit`	For temporary failures, the SAP Ariba integration toolkit retries the download. The number of retries can be specified as a value for this option. The default value is 3. Example: `retryLimit=3`
`waitTimeBetweenRetry`	The wait time between the SAP Ariba integration toolkit retries in milliseconds. For any server temporary failures, the SAP Ariba integration toolkit waits between retries.The default value is 5 minutes (300000 milliseconds). Example: `waitTimeBetweenRetry=120000`
`wsIncludeDeletedWorkspaces`	This controls whether deleted contract workspaces are exported with the Contract Workspace Export feature. To enable the parameter, set the value to **True**. By default, the parameter is disabled and set to **False**.
`exportCLIDAsCSVZip`	This controls whether the Contract Line Item Document should be exported as Excel or a CSV Zip. To export CLID as CSV Zip, set the value of the parameter to **True**. By default, the parameter is set to **False** and CLID is exported as Excel.

Combining options

The SAP Ariba integration toolkit supports the following combinations for export options:

Export Option	Valid Combination
`wsContractType, wsDateFieldName, wsDateFrom, wsDateTo, wsStatus, wsOrgId`	• `wsContractType + wsDateFieldName, wsDateFrom, wsDateTo` • `wsContractType + wsStatus` • `wsContractType + wsOrgId` • `wsContractType + wsDateFieldName, wsDateFrom, wsDateTo + wsStatus` • `wsContractType + wsDateFieldName, wsDateFrom + wsDateTo + wsOrgId` • `wsContractType + wsDateFieldName, wsDateFrom, wsDateTo + wsStatus + wsOrgId`

You cannot perform incremental exports if you specify `wsID` or `wsType` values (`wsExportType=Incremental` is ignored if you specify `wsID` or `wsType` values).

Troubleshooting errors when exporting contract workspaces

This section contains possible solutions for problems you might encounter when exporting contract workspaces.

Error	Meaning and Resolution
`Export workspace requests exceeds the limit of WSExportMaxRequests set for the period for the realm: siteName`	Ensure your workspace requests do not exceed the limit set for a realm. The total number of SAP Ariba integration toolkit workspace requests allowed per month is set in the following parameter: `Application.ACM.ITK.WSExportMaxRequestsAllowed` See Site Configuration Options for Contract Workspace Exports [page 45].
`No workspace IDs list to process.`	The SAP Ariba integration toolkit options in the `.bat` or `.sh` file does not return any result from the server. Furthermore, workspace IDs are not listed for processing. Provide the correct workspace IDs to resolve this error.
`Number of OrgIds exceed the limit of 1000.`	The `wsOrgId` field contains more than 1000 organization IDs. Reset the number of IDs specified in the `wsOrgIds` option to less than 1000 and rerun the SAP Ariba integration toolkit.
`Rerun request could not proceed as the last run was successful. You can only rerun when previous request is not successful. Please reset rerun option to false in options file and run ITK.`	The `rerun` option is set to `true` and the previous export was successful. Reset the `rerun` option to false and run the SAP Ariba integration toolkit again.

Error	Meaning and Resolution
`Some workspaces are skipped as workspace zip file preparation failed at server end. Skipped workspaces can be found at path/ skippedworkspaces.txt. Please contact customer support with skippedworkspaces.txt for further investigation.`	The SAP Ariba integration toolkit was unable to create a ZIP file for one or more workspaces. Have your Designated Support Contact log a service request with SAP Ariba. Provide the skippedworkspaces.txt file and your options file.
`There is another export workspace request is in progress for the realm: realmName. Please try after ExportRequestTimeoutInMin mins.`	Another export workspace request was started within the timeout interval. The timeout interval is specified in the following parameter: `Application.ACM.ITK.ExportRequestTimeoutInMin` See Site configuration options for contract workspace exports [page 45].
`Unable to create a workspace file.`	A common cause of this error is when the server or the response from the server is slow. You can retry rerunning the SAP Ariba integration toolkit after some time. If the problem persists, have your Designated Support Contact log a service request with SAP Ariba.
`Unable to merge the chunk files to prepare zipped files for the workspaceId.`	For larger workspace ZIP files, the SAP Ariba integration toolkit client fetches the data in smaller chunks. After the zipped files are exported, the SAP Ariba integration toolkit merges them to make a single ZIP file. An error occurred when attempting to merge the files. Rerun the SAP Ariba integration toolkit. If the problem persists, have your Designated Support Contact log a service request with SAP Ariba.

`Unable to proceed with rerun as workspace related options are changed.`
`Previous: previousRequestOptions`
`Current: currentRequestOptions`
`Please change rerun option to false and run ITK for new export request.`
	The `rerun` option is set to `true` and the previous workspace export operation did not complete, but the workspace filters in the current options file do not match the filters in the options file for the incomplete export operation.
	If you want to resume the incomplete export, set the workspace filters to the previous values. If you want to start a new export, reset the `rerun` option to false and run the SAP Ariba integration toolkit again.

Error	Meaning and Resolution
`Unable to process export request as the workspace list is not found.`	The SAP Ariba integration toolkit client retrieves a list of workspaces to be exported from the server based on the search filters in the options file. This error occurs when the workspace list is not present on the client system. This error may occur when: • the server is slow • the client system is slow and unresponsive • search filters do not display any results Verify the accuracy of inputs provided for the parameters in the options file.
`Unable to read the workspace list file.`	The SAP Ariba integration toolkit client retrieves a list of workspaces to be exported from the server based on the search filters in the options file. This error occurs if the workspace file is corrupted on the client system. Resolve any file level issues on the client system and rerun the SAP Ariba integration toolkit.
`Unable to read the workspace ZIP file for workspaceId.`	The SAP Ariba integration toolkit attempts to access the workspace data in the ZIP format from the server for each workspace ID on the list. This error occurs if the workspace ZIP file is not present on the client system. A common cause of this error is when the server is slow or if there is not enough free space on the client system. You can try accessing the workspace data after sometime or ensure there is enough free space for the workspace data to load.
`Unable to set request params stackTrace`	The SAP Ariba integration toolkit attempts to set parameters to request the server for data based on the configuration in the options file, but it is unable to set the requested parameters. Reconfigure the parameters as recommended in the options file. See Configuring the options file [page 36].
`Unable to write skipped workspaces file for this request.`	The SAP Ariba integration toolkit was unable to create a ZIP file for one or more workspaces and could not record the IDs of the skipped workspaces in the skippedworkspaces.txt file. Have your Designated Support Contact log a service request with SAP Ariba.
`Workspace search filter parameters are missing. Unable to proceed.`	Specify at least on workspace filter (`wsnnn` option, such as `wsStatus`) and rerun the SAP Ariba integration toolkit.

Site configuration options for contract workspace exports

The following configuration options control the functionality of exporting contract workspaces:

Application.ACM.ITK.ExportRequestTimeoutInMin

This parameter specifies the timeout interval, in minutes, between contract export requests from the same customer site. This interval is only checked if there is another contract export request that has not completed (the other request is in progress or was terminated by the client but has not yet terminated on the server).

> **Note**
>
> This parameter is for internal purposes only and must be changed with the assistance of Ariba Technical Support.

Default: 15.

Application.ACM.ITK.WSExportMaxRequestsAllowed

This parameter specifies the maximum number of contract export requests allowed for a customer site in a month.

Default: 2.

Application.ACM.ITK.WSExportTempFileMaxSizeInMB

This parameter specifies the maximum size of the exported workspace ZIP file.

Default: 1228 MB (1.2 GB).

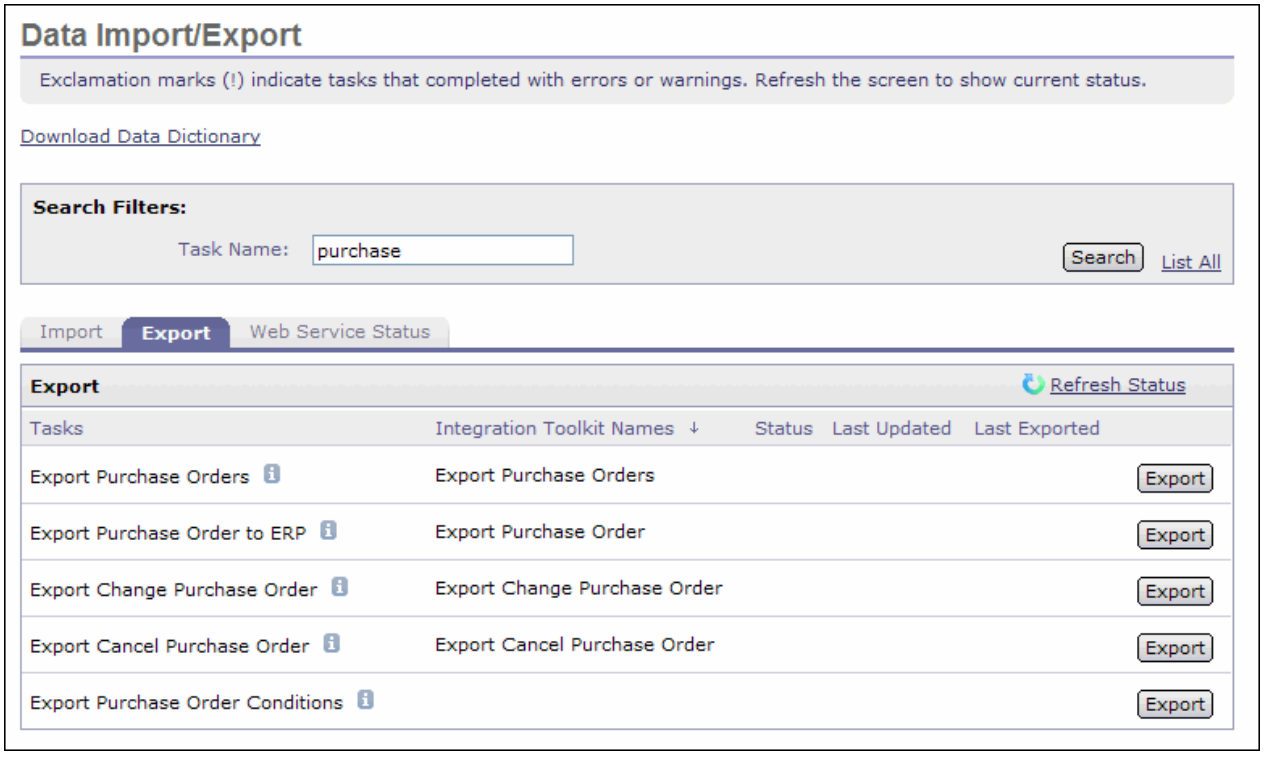
Export data format

Currently, the data export of the template consists of XML files meant only for internal purposes. With this feature, the XML files are enhanced for sharing, readability, and ease-of-use.

> **Note**
>
> See the *SAP Ariba Contracts* for a detailed description of the exported data when the Export Workspace Data event is used with the SAP Ariba data transfer tool.

Configuring events for use with the SAP Ariba data transfer tool

The events that are preconfigured to be used with the SAP Ariba data transfer tool have an SAP Ariba integration toolkit name that you must use in the options file. The SAP Ariba integration toolkit names for these events are available in the Data Import/Export page in Ariba Administrator as illustrated in the graphic below:



To upload any of the above master data, you must add a `set event` statement with the event name in the option file. This is applicable for Windows. For Unix scripts, the word SET is not required. For example, you can add the following entry to a `masterdata.bat` file:

```
set event = "Import Departments"
```

Then, the user needs to create a zip file that contains any one of the files specified below. For instance, if a user wants to upload Commodity code and Enterprise user, the zip file must contain 2 CSV files, `CommodityCode.csv` and `CustomSharedUser.csv`. To upload all master data, the user must include all the file names specified below as part of the ZIP file. The file names must match the file names that are listed as follows:

- `CommodityCode.csv`
- `ClassificationCodeMap.csv`
- `CustomSharedUser.csv`
- `CurrencyConversionRate.csv`
- `Departments.csv`
- `Regions.CSV`
- `UnitsOfMeasure.csv`
- `Products.csv`
- `SupplierOrganization.csv`

- `SupplierOrganizationOrganizationIDPart.csv`
- `SupplierParentOrganizationImport.csv`
- `SupplierProfilesImport.csv`
- `OrganizationsPullData.csv`

To include Flex Master Data (FMD) in a ZIP file for a master data upload, include a file with the name *`fmd_template_name`*`.csv`, where *`fmd_template_name`* is the name of your FMD template in simplified form (the name with spaces removed).

All other events must be defined before you can use them with the SAP Ariba data transfer tool, and single events must be included in a batch event if they have not been included in any other existing batch events.

SAP Ariba data transfer tool syntax

The main program used to run the SAP Ariba data transfer tool is:

- `aribafiletransfer.bat` (Microsoft Windows)
- `aribafiletransfer.sh` (UNIX)

The main program takes an options file, which sets the environment variables, as an argument. The main program then continues, invoking the appropriate wrapper scripts as necessary to perform the requested data transfer task.

The basic SAP Ariba data transfer tool program syntax is:

`aribafiletransfer.bat data-download/optionFile.bat` (Microsoft Windows)

`aribafiletransfer.sh data-download/optionFile.sh` (UNIX)

The SAP Ariba data transfer tool program syntax for exporting SAP Ariba Spend Analysis data is:

`aribafiletransfer.bat sv-download\optionsFile.bat` (Microsoft Windows)

`aribafiletransfer.sh sv-download/optionFile.sh` (UNIX)

Before you run the tool, ensure that you have set up the appropriate options files for the tasks you plan to run.

Enhanced monitoring of the SAP Ariba integration toolkit

When a SAP Ariba integration toolkit run is successfully completed, it sends an email notification indicating the status of the data transfer to the email address configured as part of the options file. This email notification provides only high-level information, such as data transfer successful or data transfer failed. It does not provide any information about the status of the event run or the number of records that were created, updated, deleted, or failed as a result of errors.

The enhanced monitoring of the SAP Ariba integration toolkit ensures that information about the event run and the number of records affected is captured and sent as email notifications to the desired recipients within your organization. These email notifications provide detailed information such as:

- Name of the batch import integration event
- Name of the individual import integration event
- Status of the batch or individual import integration event
- Name of the customer site for which the integration events are being run
- Date / Time when the event processing started and ended
- The number of records that are created, updated, or deleted
- The number of records that failed as a result of errors

> **Note**
>
> This functionality is available only for import events, because the information in the email notification is useful primarily when you import data from your ERP system into the Ariba system.

Types of email notifications

When you use the SAP Ariba integration toolkit to run a batch import event, two types of email notifications are sent based on the status of the integration event: Submitted Summary and Processed Summary. The Submitted Summary is sent when the SAP Ariba server receives the integration event request. The Processed Summary is sent when the SAP Ariba server has completed the execution of the event.

The Submitted Summary and Processed Summary email notifications are sent for the batch import event and also for each import integration event included in the batch. Therefore, if you have included five integration events within a batch import event, then the recipients receive 12 email notifications. One Submitted Summary email notification for the batch import event and five Submitted Summary email notifications for each import integration event within the batch. Similarly, One Processed Summary email notification for the batch import event and five Processed Summary email notifications for each import integration event within the batch.

Subject line of the email notifications

The subject line for both Submitted Summary and Processed Summary email notifications is different. The subject line of the Submitted Summary email notification both at the batch level and individual integration event level is constructed based on the following parameters:

```
ITK Event[Name of the Integration Event or Name of the Batch Integration
Event];Status[Status of the Integration Event];Site[Name of the Customer Site]
```

The subject line of the Processed Summary email notification both at the batch level and individual integration event level is constructed based on the following parameters:

```
ITK Event[Name of the Integration Event or Name of the Batch Integration
Event];Status[Status of the Integration Event];Site[Name of the Customer Site]Processed
Status[Status of the Batch Event]
```

> **Note**
>
> Based on the subject lines of these notifications, the recipients of these notifications can configure rules in their respective email clients to efficiently manage these notifications.

Sample email notifications

Sample email notifications sent for the Submitted Summary and Processed Summary at the batch level and individual integration event level are displayed as follows:

Submitted summary email notification at the batch level

```
From: notificationAdmin@your.admins.domain
Subject:ITK Event[SupplierProfileBatchPull];Status[Submitted];Site[mycoolcompany];
Integration Event Name: SupplierProfileBatchPull
Status: Submitted
Site: mycoolcompany
```

Submitted summary email notification at the individual integration event level

```
From: notificationAdmin@your.admins.domain
Subject:ITK Event[SupplierProfileFileImport];Status[Submitted];Site[mycoolcompany];
Integration Event Name: SupplierProfileFileImport
Status: Submitted
Site: mycoolcompany
```

Processed summary email notification at the batch level

```
From: notificationAdmin@your.admins.domain
Subject: ITK Event[SupplierProfileBatchPull];Status[Completed
(*)];Site[mycoolcompany];Processed Status[Completed (*)];
Integration Event Name: SupplierProfileBatchPull
Status: Completed (*)
Site: mycoolcompany
Date/Time Event Processing Started:6/7/2012 2:50 AM
Date/Time Event Processing Ended:6/7/2012 2:50 AM
Summary:
Records Created:0
Records Deleted:0
Records Updated:77
Error Messages:1
```

Processed summary email notification at the individual integration event level

```
From: notificationAdmin@your.admins.domain
Subject: ITK Event[SupplierProfileFileImport];Status[Completed
(*)];Site[mycoolcompany];Processed Status[Completed (*)];
Integration Event Name: SupplierProfileFileImport
Status: Completed (*)
Site:mycoolcompany
Date/Time Event Processing Started:6/7/2012 2:50 AM
Date/Time Event Processing Ended:6/7/2012 2:50 AM
Summary:
Records Created:0
Records Deleted:0
Records Updated:77
Error Messages:1
```

Note

- In the subject line of the Processed Summary email notification, if the `Status` and `Processed Status` fields contain the value `Completed (*)`, it indicates that some of the records have failed as a result of errors.
- The content of these email notifications is available in the English language only. In addition, the integration event name contained in the notification is the display name of the integration event and not the SAP Ariba integration toolkit name.

Configuring email addresses

To ensure that Submitted Summary and Processed Summary email notifications are sent to the desired recipients within your organization, your Administrator must specify the email addresses of the recipients by using the Ariba Administrator of your respective SAP Ariba solution.

1. In Ariba Administrator, under **Integration Manager** workspace, click **Integration Toolkit Security**.

   ### Note

   You must be a member of the Integration Administrator group to access this page in Ariba Administrator. This page also allows the Administrator to specify the authentication method that is used to authenticate data upload and download requests from the SAP Ariba integration toolkit. The administrator can select authentication based on Shared Secret or Digital Certificate.

2. Under **Notification**, in the **Email Addresses** field, enter the required email addresses.

   ### Note

   You can specify a maximum of five email addresses separated by a comma (,) or semicolon (;). The maximum limit of this field is 2048 characters. If you specify an email address that does not exist, then no Undelivered email notification is sent back to the Ariba Server.

3. Click **Save**.

Uploading data

The SAP Ariba data transfer tool uploads data in three modes:

- **Full master data load** uses the default operation that is defined for each integration task in a batch. The default operation is different for each event in the batch you are uploading.
  For example, for "Load and Delete" events, objects that are not specified in the data are "deleted" (that is, deactivated). In this case, if you upload `user.csv`, then all records are updated to match the current version of the file. If you upload only part of `user.csv`, then only the records listed in the partial file are created or updated. All other records are deactivated.
- **Incremental master data load** provides a safe transfer mode—no objects are deactivated. Specifically, "Load and Delete" events are actually run as just "Load" operations.
  For example, if you want to add new suppliers, you can generate a partial `supplier.csv`, run an incremental master data upload, and update a few records. In this case, records not listed in the partial file are not deactivated.
- **Transactional data uploads**, where the tool creates and modifies objects in the database using values in the data file. These uploads are also "safe" because no objects are deactivated. Note that you can upload transactional status (ID /Error) and remittance files from your back-end system. You cannot upload user, supplier, or any other types of files using transactional data upload.

> **Note**
>
> With Spend Visibility data loads, the tool uploads ZIP files as specified in the SAP Ariba Spend Analysis data acquisition schema.

For each integration task in a batch, a default operation is defined that can be any of the following:

Operation	Description
`Create`	Creates a ClusterRoot object. If the object exists in the database prior to this operation, a non-fatal exception is inserted in the integration event log entry, and the object is not created.
`Delete`	Deactivates a Cluster Root object. It does nothing, if the object does not exist in the database or is already deactivated.
`Update`	Updates an existing ClusterRoot object. If the object does not exist in the database, a non-fatal exception is inserted in the integration event entry, and the object is not updated.
`Load`	Is the combination of Create and Update. Specifically, if the object does not already exist, it is created. Otherwise, it is updated.
`Update And Delete`	Like Update, updates records in the database if the records are there in the file and also there in the database. This option also does the following: • Deletes records from the Ariba database that have the same EventSource label as the integration event and are not there in the file. • Does not create new elements, even if they exist in the data source file.

Operation	Description
`Load And Delete`	• Creates records in the database if the records are there in the file and not there in the database. • Updates records in the database if the records are there in the file and also there in the database. • Deletes records in the database if the records are not there in the file. However, it will only delete records in the database, where their adapter source matches the event source of the Load and Delete event being run.
`Update Elements Only`	Updates the contents of vector fields. Mostly used in header detail imports. Specifically, for a given vector field, if the vector contains the object, the object is updated. Otherwise, the updated object is inserted into the vector.
`Delete Elements Only`	Deletes elements from vector fields. Mostly used in header/detail imports. Specifically, removes the object from the vector field if the vector contains the object.

Related Information

Automated extended validation [page 59]

File size limits for data upload requests

An upload request sent by the SAP Ariba data transfer tool to the SAP Ariba system can contain ZIP files and/or individual CSV files. The total size of files within an upload request sent by SAP Ariba data transfer tool to the SAP Ariba system must be less than 10 MB. This is applicable for Full Master Data, Incremental Master Data, and Transactional Data uploads. However, if you run an upload event through the Ariba Administrator, then the size of the CSV/ZIP file must be less than 20 MB.

Data upload examples

The following example shows the minimum options that you must configure to use the SAP Ariba data transfer tool. With this minimum configuration, the tool can upload data to your SAP Ariba system, send email notification regarding the data transfer status, and clean up data files that are older than seven days.

> **Note**
>
> This example assumes you are running the tool on Microsoft Windows. To use the tool on UNIX, use the `.sh` options files and forward slashes (`/`) instead, and follow the syntax as it appears in the options files.

About full master data upload

This example explains how to use the tool to complete a full master data upload, and assumes that the following are true:

- You have obtained your shared secret or digital certificate. If you have not yet encrypted the shared secret, you do so in step 2.
- You have obtained the site name for your company as defined in your site URL.
- Your Ariba service is up and running.
- Java 1.5 or 1.6 is installed in `C:\Java`.

> **Note**
>
> This example shows the Microsoft Windows `set` command. The UNIX options files do not require you to use the `set` command.

Example: Uploading full master data

Procedure

1. Make a copy of `D:\IntegrationToolkit\sample-options\DataTransferTool\data-upload\masterdata.bat.sample`, move the copy to another directory, and rename the file `masterdata.bat`.

   > **Note**
   >
   > On Microsoft Windows, the file extension must be `.bat`. (This does not apply if you are using UNIX.)

2. If you are using shared secret based authentication, then you must encrypt the customer key and shared secret. See About security encryption [page 24].
3. You can now modify `masterdata.bat`.
4. Provide values for the following options in `masterdata.bat`:
   a. Define the `site` option:

      ```
      set site=mycoolcompany
      ```

      where `mycoolcompany` is the unique name of the company as defined in your URL. For example, in the URL `http://mycoolcompany.procurement.ariba.com`, the `site` is `mycoolcompany`.
   b. Enter your encrypted shared secret.
   c. Define the `event` option to specify the master data load event appropriate for your SAP Ariba solution. See Configuring events for use with the SAP Ariba data transfer tool [page 46].

      For example, if you are importing master data into SAP Ariba Sourcing, use the `Import Master Data` event. If you are importing data into SAP Ariba Procurement solutions, use the `Import Batch Data` event.

d. Define the `topDir` option to specify the top level directory in which your back-end system places data for transfer:

```
set topDir=D:\ariba\DataTransferTool\topLevelDir
```

You can use any directory as `topLevelDir`, as long as it contains an empty subdirectory called `InDir`. The tool assumes `topLevelDir` and `InDir` exist.

> **Note**
>
> If you are using the SAP Ariba data transfer tool in an SAP environment, the `inDir` value must match a value in SAP. See the *Integrating SAP Ariba cloud solutions procurement and invoicing data with SAP ERP and SAP S/4HANA* for more information on using `inDir`.

e. Define the `notifyEmail` option to specify the email address to which notifications are sent from the tool:

```
set notifyEmail=your email account
```

> **Note**
>
> You can specify multiple email addresses by separating each one with a comma.

f. Define the `smtpMailServer` option to specify the SMTP mail server to use for email notifications from the tool:

```
set smtpMailServer=mail.yourcompany.com
```

g. Specify the `JAVA_HOME` variable:

```
set JAVA_HOME=C:\Program Files\Java\jre_version\
```

For UNIX, set Java Home as `export JAVA_HOME=/usr/local/java/jre_version`

5. Save your changes.
6. The following sub steps explain the behavior when the file to be uploaded is not available in the `InDir`:

> **Note**
>
> Step 7 explains how to move the file to be uploaded to the `InDir` and the behavior thereafter.

a. To open a Microsoft Windows command interpreter, click **Start**, **Run**, then type `cmd` and click **OK**.
b. Enter:

```
cd D:\ariba\DataTransferTool
```

and press Return.

c. Enter:

```
bin\aribafiletransfer directoryPath\masterdata.bat
```

and press Return. (Where *directoryPath* is the directory path to your `masterdata.bat` options file.)

d. Check the command interpreter for a message similar to the one below, and also check your incoming email for a message about the transfer.

```
Will log to D:\ariba\DataTransferTool\topLevelDir\logs
\log-20061012-93835-85.log
*****************************************************
****Running File Transfer Tool at Thu Oct 12 09:38:36 PDT 2006****
no input items to process, exiting..
Data transfer complete.
Sending D:\ariba\DataTransferTool\topLevelDir\logs\log-20061012-93835-85.log
to joesmith@ariba.com
Subject: AribaFileTransfer Master Data Upload: Successful
Login name is donotreply
```

e. Open the log file.

The log entry `no input items to process, exiting` indicates that the tool did not upload any data (in this case, because there are no files in `InDir` yet).

7. Upload data as follows:

a. Locate your master copy of a CSV file for uploading, such as `Address.csv`. As this is a full master data upload, you must have all addresses that you need in the file. Any address objects in the SAP Ariba database that do not have a corresponding entry in `Address.csv` file are deactivated. This is a "Load and Delete" operation.

b. Copy `Address.csv` to `D:\ariba\DataTransferTool\topLevelDir\InDir\batch0`.

> **Note**
>
> Any directory that does not begin with the word `processing` works. See the `filterPrefix` option in Troubleshooting data transfers [page 66].

c. Enter:

```
bin\aribafiletransfer directoryname\masterdata.bat
```

and press Return. (Where *directoryPath* is the directory path to your `masterdata.bat` options file.)

d. Check the command interpreter for a message similar to the one below, and also check your incoming email for a message about the transfer.

```
Will log to D:\ariba\DataTransferTool\topLevelDir\logs
\log-20061012-154159-61.log
*****************************************************
****Running File Transfer Tool at Thu Oct 12 15:42:01 PDT 2006****
done executing POST method.
moving D:\ariba\DataTransferTool\topLevelDir\InDir\batch0 to D:
\DataTransferTool\topLevelDir\OutDir\success\batch0.20061012154202
Data transferr complete.
Sending D:\ariba\DataTransferTool\topLevelDir\logs\log-20061012-154159-61.log
to joesmith@ariba.com
Subject: AribaFileTransfer Master Data Upload: Successful
Login name is donotreply
```

Results

You have successfully uploaded a master data file. To verify this data transfer, check with your customer administrator. (Your administrator can check the Import Batch Data import event in Ariba Administrator. If the sample data was transferred successfully, the event must have a status of `Complete`.)

About this example

By defining the `topDir` option, you automatically specify the following directories:

- By default, the log file is in a subdirectory called `logs` under `topDir`.
- By default, the tool transfers CSV files from `%topDir%\InDir\directoryname`, where `directoryname` is any directory that does not have the prefix `processing`. The tool transfers all `directoryname\*.csv` files.
- By default, the tool moves the original CSV file to the directory `%topDir%\OutDir\success \directoryname.YYYYMMDDhhmmss` once the file is successfully transferred. In this example, `directoryname` is `batch0`, and `YYYYMMDDhhmmss` is the time stamp `20061012154202`.
- By default (not shown in this example), a permanent failure on the server causes the tool to move the original CSV file to the directory `%topDir%\OutDir\failed\directoryname.YYYYMMDDhhmmss`.
- By default (not shown in this example), any CSV files that the tool does not transfer are moved to the directory `%topDir%\UnSent\directoryname.YYYYMMDDhhmmss`. This can happen if you have more than one set of batch data (for example, `batch0`, `batch1`, and `batch2`) and you are doing a full master data upload. In this operation, the tool transfers only the latest set of batch data.

About full and incremental data loads

Both types of master data load, create or update objects in the database using the values in the data files. For a full data load, all data is loaded. If an object in the database does not exist in a data file, it is deactivated in the database.

> **Note**
>
> If a data file (ZIP file) includes a CSV file that contains only column headers and no records, the Full data Load does not deactivate all the data. For example, if the ZIP file includes a TaxCode.csv file that contains only the character encoding row (such as “UTF-8”) and the column header row (such as "SETID_TAX_CD_,TAX_CD,SETID,TAX_CD_DESCR_,DESCR"), Full Load does not deactivate all tax codes. However, if the file contains a single tax code record, Full Load adds or updates that tax code and deactivates all other tax codes.

An incremental data load does not necessarily load all data. Therefore, if an object in the database does not exist in a data file, it is **not** deactivated in the database.

In all other respects the data loads are the same. If an object in a data file:

- Does not already exist in the database, it is created.
- Already exists in the database, it is updated using the value in the data file.

- Already exists in the database and the object was previously deactivated, it is reactivated.

Note that some data files may update objects that were previously created by other data files in the ZIP file.

File and directory handling

This section describes how the tool transfers batch data files during data uploads.

Full Uploads

For full master data loads, the SAP Ariba data transfer tool retrieves and transfers only the latest batch of CSV files. If there are two batches of full-load CSV files, the tool compares the creation dates of their subdirectories and moves the older one to the `%UnSentDir%` directory.

After transferring the new files, the tool adds a time stamp to the subdirectory that contained them and moves the subdirectory to the `%outDir%\success` directory.

If the tool is unable to transfer the files, such as if a remote permanent failure occurs, the tool moves the files to the `%outDir%\failed` directory.

Incremental Uploads

The incremental master data upload and all transaction uploads are incremental. That means each batch of CSV files only contains data that changed since the last upload. Instead of transferring only the latest subdirectory, the tool transfers all of the CSV subdirectories, from the oldest to the newest. The tool stops when it completes the transfer or encounters an error. There is no need for an unsentDir directory.

During incremental upload, the SAP Ariba data transfer tool transfers data from the `InDIR` in a sequence. This ensures that after the SAP Ariba data transfer tool transfers data from the first folder to SAP Ariba Procurement solutions, it starts to move the data from the next folder. This process is repeated till all the data in the `InDIR` is processed. However, when the SAP Ariba data transfer tool starts to move the data from a folder, if data from the previous folder is still in processing in SAP Ariba Procurement solutions, a server busy is reported. To ensure that such an error does not occur, you can use the `waitTimeBetweenIncUpload` parameter. The value for this parameter must be set it milliseconds. If you have set this parameter in the batch file, the SAP Ariba data transfer tool after sending data from a particular folder, will wait for the configured time interval before it starts to process the data from the next folder.

After the upload, the tool adds a time stamp to the subdirectory that contained them and moves the subdirectory to the `%outDir%\success` directory.

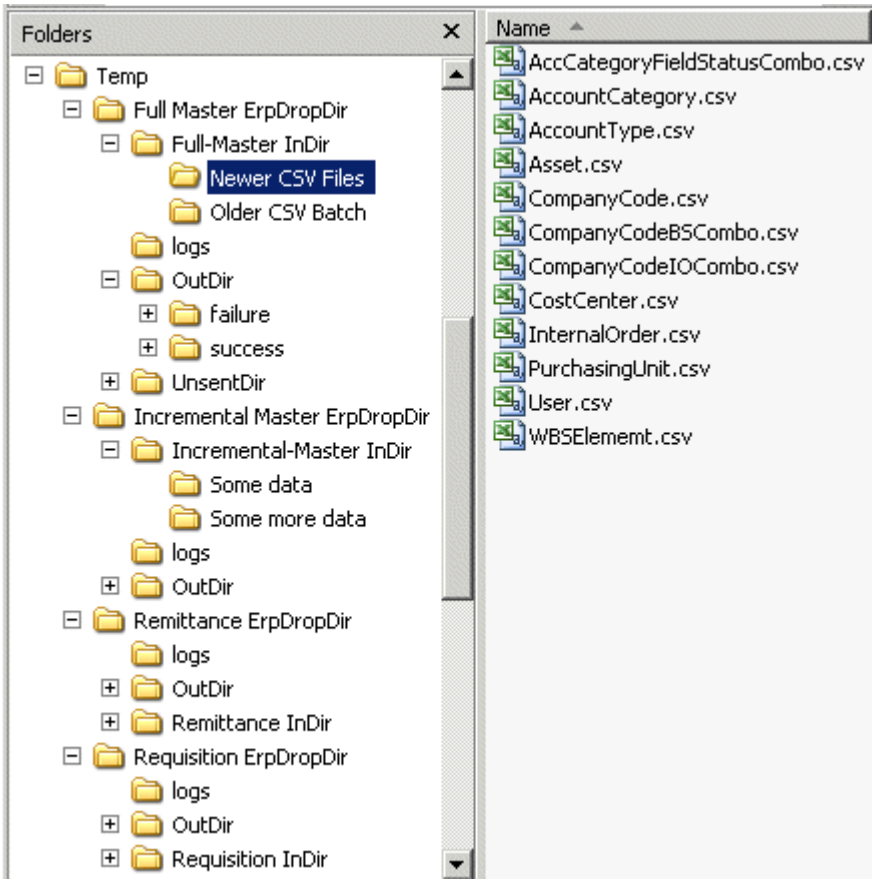
If the tool is unable to transfer the files, such as if a connection failure occurs, the tool moves the files to the `%outDir%\failed` directory.

All Uploads

If you use a utility to create CSV files from your back-end system data, it must name the batch subdirectory with a special prefix, such as `processing`, while the file is being written. The `filterPrefix` option (whose default is `processing`) prevents the tool from transferring CSV files with that prefix. The utility used to create the files can then rename the directory when the files are complete. Using the `filterPrefix` variable prevents errors in the case where the upload is scheduled to run when the CSV files are still being written.

You can use the `performCleanup` options to clean old files out of the outDir and unsentDir directories.

The figure below shows an example of a data structure set up for different upload tasks. Each task is separated into a different top directory to keep the log files, and successful/failed data files separate from those of other tasks.



For the next scheduled full master data upload, only the `Newer CSV Files` directory (highlighted) is uploaded. It will then be moved to the OutDir directory and have a time stamp appended, such as `Newer CSV Files.yyyymmddhhmmss`. The directory called `Older CSV Files` is moved to the `UnsentDir` directory.

Just before this image was taken, the CSV directory might have been called `Processing Newer CSV Files`. Had the upload task run at that time, the directory might have been ignored and `Older CSV Files` might have been uploaded.

For the next incremental master data upload, both subdirectories, `Some data` and `Some more data`, are uploaded.

Automated extended validation

Program Manager in SAP Ariba Spend Analysis can perform automated extended validation while uploading files from SAP ERP using the SAP Ariba integration toolkit on SAP Netweaver PI to:

- **Perform duplicate row check**: Checks for duplicate rows.
- **Perform dimension reference check and Invoice to PO match**: Checks for data integrity between the invoice files and the associated master data files, such as, supplier master data, accounts master data, cost center master data, and so on.

> **Note**
>
> The file to be uploaded must be either a character-delimited file or a ZIP file.

Additional references

For more information about the extended validations in SAP Ariba Spend Analysis, see the Uploading Data Files section in the *Ariba Reporting and Analysis Data Load and Administration Guide*.

Downloading data

Depending on your SAP Ariba solution, when you download data from your SAP Ariba system, the SAP Ariba data transfer tool triggers one of the following tasks:

- Export Expense Reports
- Export Payment Requests in 3 File Format (for Generic only)
- Export Payment Requests
- Export Accrual Transactions (note that you must use the string "`Export Accrual Transactions`" in the `event` option in the SAP Ariba data transfer tool options file)
- ExportSchema (for SAP Ariba Spend Analysis only)

These tasks send selected CSV files from your SAP Ariba system to the directory you specify in the associated options file. After you download data, you can use any utilities you may have to extract the data from the CSV files for transfer to your back-end system.

Downloading master data

The SAP Ariba integration toolkit only supported export of transactional data which is on the PendingDataQueue.

Now, the integration toolkit supports both master and transactional data exports.

Data download examples

This example shows the minimum options that you must configure to use the SAP Ariba data transfer tool to download data. With this minimum configuration, the tool can download data from your SAP Ariba system, send email notification regarding the data transfer status, and clean up data files that are older than seven days.

> **Note**
>
> This example assumes you are running the tool on Microsoft Windows. If you use the tool on UNIX, use the `.sh` options files and forward slashes (`/`) instead, and follow the syntax as it appears in the options files.

This example assumes that the following are true:

- You know your shared secret or digital certificate.
- You have obtained the site name for your company as defined in your site URL.
- Your Ariba service is up and running.
- Java 1.5 or 1.6 is installed in `C:\Java`.

> **Note**
>
> This example shows the Microsoft Windows `set` command. The UNIX options files do not require you to use `set`.

Example: Downloading transactional data

Context

This example shows how to download expense reports.

> **Note**
>
> This example is for illustrative purposes only. As a best practice, to download more than one type of transactional data, use a separate options file instead of modifying the same file. That way you can specify different options, such as a different download directory.

Procedure

1. Make a copy of `D:\ariba\sample-options\DataTransferTool\data-download\transactionaldata.bat.sample`, move the copy to another directory, and rename the file `transactionaldata.bat`.

   > **Note**
   >
   > On Microsoft Windows, the file extension must be `.bat`. (This does not apply if you are using UNIX.)

2. If you are using shared secret based authentication, then you must encrypt the customer key and shared secret. See About security encryption [page 24].
3. Provide values for the following options in `transactionaldata.bat`.
   a. Define the `site` option:

      ```
      set site=mycoolcompany
      ```

      where `mycoolcompany` is the unique name of the company as defined in your URL. For example, in the URL `http://mycoolcompany.procurement.ariba.com`, the `site` is `mycoolcompany`.
   b. Enter your encrypted shared secret.
   c. Define the `export` event option as appropriate for your SAP Ariba solution.
   d. Define the `downloadDir` option to specify the directory into which you want to download data from Ariba:

      ```
      set downloadDir=D:\ariba\DataTransferTool\DownloadFromAriba
      ```

   e. Define the `notifyEmail` option to specify the email address to use for notifications from the tool:

      ```
      set notifyEmail=your email account
      ```

      > **Note**
      >
      > You can specify multiple email addresses by separating each one with a comma.

   f. Define the `smtpMailServer` option to specify the SMTP mail server to use for email notifications from the tool:

      ```
      set smtpMailServer=mail.yourcompany.com
      ```

   g. Specify the `JAVA_HOME` variable:

      ```
      set JAVA_HOME=C:\Program Files\Java\<jre version>\
      ```

      For UNIX, set Java Home as `export JAVA_HOME=/usr/local/java/`<jre version>
4. Save your changes.
5. Download the expense reports as follows:
   a. To open a Microsoft Windows command interpreter, click **Start**, **Run**, then type `cmd` and click **OK**.
   b. Enter:

      ```
      cd D:\ariba\DataTransferTool
      ```

      and press Return. (`D:\ariba\DataTransferTool` is the directory in which the tool is installed.)
   c. Enter:

      ```
      bin\aribafiletransfer sample-options\data-download\transactionaldata.bat
      ```

      and press Return.
   d. Check the command interpreter for a message similar to the one below, and also check your incoming email for a message about the transfer.

      ```
      Will log to D:\ariba\DataTransferTool\DownloadFromAriba\logs
      \log-20061012-193554-46.log
      ************************************************
      ****Running File Transfer Tool at Thu Oct 12 19:35:55 PDT 2006****
      ```

```
done executing POST method.
unzipping D:\ariba\DataTransferTool\DownloadFromAriba
\~tmp49582\~download45181.tmp to
D:\ariba\DataTransferTool\DownloadFromAriba\~tmp49582\unzip45180
D:\ariba\DataTransferTool\DownloadFromAriba\~tmp49582\~download45181.tmp
successfully unzipped Reponse successfully downloaded to the directory 'D:
\ariba\DataTransferTool\DownloadFromAriba\Expense20061012193557'
Data transfer complete.
```

Example: Downloading payments

This example shows how to download payments.

1. Open `transactionaldata.bat`.
2. Specify the `event` option:

   ```
   set event="Export Payment Requests"
   ```

   The default is `Export Expense Reports`.

   > **Caution**
   >
   > On Microsoft Windows, quotes are required for the `event` option.

3. Specify the `filePrefix` option:

   ```
   set filePrefix=Payment
   ```

   The default is `Expense`.
4. Specify a different time stamp file using the `timestampFile` option:

   ```
   set timestampFile=%downloadDir%\PaymentTimeStamp
   ```

> **Note**
>
> If you are using the SAP Ariba data transfer tool in an SAP environment, the `filePrefix` value must match a value in SAP. See the *Integrating SAP Ariba cloud solutions procurement and invoicing data with SAP ERP and SAP S/4HANA* for more information on using `filePrefix`.

About This Example

Note the following about this download example:

- The log file is automatically created under `%downloadDir%\logs`.
- The tool creates a `tmp` directory (in this case, `~tmp49582`) to receive the data from the server.
- When the tool finishes transferring the CSV files (there are three in this example), the tool renames the temp directory to `D:\ariba\DataTransferTool\DownloadFromAriba\Expense20061012193557`, which is `%downloadDir%\%filePrefix%`*`YYYYMMDDhhmmss`*

- Open the contents of `D:\ariba\DataTransferTool\DownloadFromAriba`, and note the following:
  An empty file named `ExpenseTimeStamp` is created with a time stamp corresponding to `20061012193557`. The next time you run the tool, it uses this time stamp to determine the last successful download time. The tool looks only for data with a more recent time stamp than this to download incrementally.
  To download incremental data successfully, you must specify the same time stamp file for this event (unless there is good reason not to do so, such as if you are troubleshooting). By default, the time stamp file is named `ExpenseTimeStamp`, which is the default name even for payment exports. You must specify a different value for the `timestampFile` option if you are doing a payment or accrual export.

In this example, the tool transferred the following three files from the SAP Ariba application into `D:\ariba\DataTransferTool\DownloadFromAriba\Expense20061012193557`. If there is no expense report data to be exported from the SAP Ariba application, these files are empty.

- `ExpenseReportHeaderExport.csv`
- `ExpenseReportDetailExport.csv`
- `ExpenseReportDistributionExport.csv`

Directory structure

> **Note**
>
> This section does not apply to SAP Ariba Spend Analysis.

The SAP Ariba data transfer tool places data downloaded from the SAP Ariba solution in the directory specified in the `downloadDir` option. By default, this directory is called `C:\TEMP\DownloadFromAriba`.

Data download examples for SAP Ariba Sourcing

This example shows the minimum options that you must configure to use the SAP Ariba data transfer tool to download data. With this minimum configuration, the tool can download data from your SAP Ariba system, send email notification regarding the data transfer status, and clean up data files that are older than seven days.

> **Note**
>
> This example assumes you are running the tool on Microsoft Windows. If you use the tool on UNIX, use the `.sh` options files and forward slashes (`/`) instead, and follow the syntax as it appears in the options files.

This example assumes that the following are true:

- You know your shared secret or digital certificate.
- You have obtained the site name for your company as defined in your site URL.
- Your Ariba service is up and running.
- Java 1.5 or 1.6 is installed in `C:\Java`.

> **Note**
>
> This example shows the Microsoft Windows `set` command. The UNIX options files do not require you to use the `set` command.

Example: Downloading transactional data for SAP Ariba Sourcing

Procedure

1. Make a copy of `D:\ariba\sample-options\DataTransferTool\data-download\transactionaldata.bat.sample`, move the copy to another directory, and rename the file `EnterpriseUsers.bat`.

   > **Note**
   >
   > On Microsoft Windows, the file extension must be `.bat`. (This does not apply if you are using UNIX.)

2. If you are using shared secret based authentication, then you must encrypt the customer key and shared secret. See About security encryption [page 24].
3. Provide values for the following options in `EnterpriseUsers.bat`.
   a. Define the `site` option:

      ```
      set site=mycoolcompany
      ```

      where `mycoolcompany` is the unique name of the company as defined in your URL. For example, in the URL http://mycoolcompany.sourcing.ariba.com, the `site` is `mycoolcompany`.
   b. Enter your encrypted shared secret.
   c. Define the `export` event option as `Export Enterprise Users`.
   d. Define the `downloadDir` option to specify the directory into which you want to download data from Ariba:

      ```
      set downloadDir=D:\ariba\DataTransferTool\DownloadFromAriba
      ```

   e. Define the `notifyEmail` option to specify the email address to use for notifications from the tool:

      ```
      set notifyEmail=your email account
      ```

      > **Note**
      >
      > You can specify multiple email addresses by separating each one with a comma.

   f. Define the `smtpMailServer` option to specify the SMTP mail server to use for email notifications from the tool:

      ```
      set smtpMailServer=mail.yourcompany.com
      ```

   g. Specify the `JAVA_HOME` variable:

      ```
      set JAVA_HOME=C:\Program Files\Java\<jre version>\
      ```

For UNIX, set Java Home as `export JAVA_HOME=/usr/local/java/`<jre version>

h. Set the `FilePrefix` option as required.

Set `FilePrefix= Enterprise`

4. Save your changes.
5. Download the enterprise users as follows:
   a. To open a Microsoft Windows command interpreter, click **Start**, **Run**, then type `cmd` and click **OK**.
   b. Enter:

```
cd D:\ariba\DataTransferTool
```

   and press Return. (`D:\ariba\DataTransferTool` is the directory in which the tool is installed.)
   c. Enter:

```
bin\aribafiletransfer sample-options\data-download\EnterpriseUsers.bat
```

   and press Return.
   d. Check the command interpreter for a message similar to the one below, and also check your incoming email for a message about the transfer.

```
Will log to D:\ariba\DataTransferTool\DownloadFromAriba\logs
\log-20061012-193554-46.log
**************************************************
****Running File Transfer Tool at Thu Oct 12 19:35:55 PDT 2006****
done executing POST method.
unzipping D:\ariba\DataTransferTool\DownloadFromAriba
\~tmp49583\~download45181.tmp to
D:\ariba\DataTransferTool\DownloadFromAriba\~tmp49583\unzip45180
D:\ariba\DataTransferTool\DownloadFromAriba\~tmp49583\~download45181.tmp
successfully unzipped Reponse successfully downloaded to the directory 'D:
\ariba\DataTransferTool\DownloadFromAriba\Enterprise20061012193557'
Data transfer complete.
```

> **Note**
>
> This example is for illustrative purposes only. As a best practice, if you want to download more than one type of transactional data, use a separate options file instead of modifying the same file. That way you can specify different options, such as a different download directory.

About this example

Note the following about this download example:

- The log file is automatically created under `%downloadDir%\logs`.
- The tool creates a `tmp` directory (in this case, `~tmp49583`) to receive the data from the server.
- When the tool completes the transfer of the CSV files, the tool renames the temp directory to `D:\ariba\DataTransferTool\DownloadFromAriba\Enterprise20061012193557`, which is `%downloadDir%\%filePrefix%`*`YYYYMMDDhhmmss`*
- Open the contents of `D:\ariba\DataTransferTool\DownloadFromAriba`, and note the following: An empty file named `EnterpriseTimeStamp` is created with a time stamp corresponding to `20061012193557`. The next time you run the tool, it uses this time stamp to determine the last successful

download time. The tool looks only for data with a more recent time stamp than this to download incrementally.
To download incremental data successfully, you must specify the same time stamp file for this event (unless there is good reason not to do so, such as if you are troubleshooting).

Example: Downloading another data type for SAP Ariba Sourcing

1. Open `transactionaldata.bat`.
2. Specify the `event` option:

```
set event="Export Contract Header Data (ZIP)"
```

> **Caution**
>
> On Microsoft Windows, quotes are required for the `event` option.

3. Specify the `filePrefix` option as required:

```
set filePrefix=Contract
```

4. Specify a different time stamp file using the `timestampFile` option:

```
set timestampFile=%downloadDir%\ContractTimeStamp
```

> **Note**
>
> If you are using the SAP Ariba data transfer tool in an SAP environment, the `filePrefix` value must match a value in SAP. See the *Integrating SAP Ariba cloud solutions procurement and invoicing data with SAP ERP and SAP S/4HANA* for more information on using `filePrefix`.

Troubleshooting data transfers

This section contains possible solutions for problems you might encounter while transferring data.

Problem	Solution
A failure occurs	Make sure the SAP Ariba service is currently available Check the log file for information Check the options files for incorrect parameters Check the command interpreter for incorrect or ambiguous input
A temporary failure occurs when you have a backlog of incremental data to upload and the transfer times out before completing	Let the tool complete processing the backlog. Until the backlog is processed completely, you might receive multiple email messages indicating the transfer failed. This problem is only temporary.

Problem	Solution
You receive an error message containing this text: `option must start with '-'`	Check your options file for possible incorrectly quoted options.
You receive an error message containing this text: `Bad credentials -- wrong shared secret`	The shared secret is not correct. Check the shared secret and try again.
You receive an error message indicating you tried to transfer a file with an illegal filename	Ensure that the filename is typed correctly and that it is one of the CSV files that the SAP Ariba solution expects. The CSV files expected are listed in the data load guide for each SAP Ariba solution.
You receive an error message containing this text: `Invalid value for parameter sourceSystem`	Ensure that you specify the value for the `sourceSystem` parameter.
You receive an error message containing this text: `You can configure only one of the following authentication methods, password, sharedSecret or clientKeystore`	Ensure that you use only one authentication method at a time.
You receive an error message containing this text: `input cannot be multiple zip files or a combination of zip and csv files`	Ensure that you place either a single ZIP file or CSV files, but not both, in the directory configured for SAP Ariba Spend Analysis uploads.
You want more log information	Append the `verbose` option to the `additionalClientArgs` option. For example: `set additionalClientArgs=-filterPrefix %filterPrefix% -verbose`

SAP Ariba data transfer tool error codes

The error codes that the SAP Ariba data transfer tool can return appear on the command line as either `0`, `1`, `2`, or `3`. The following table describes the error codes.

Error Code	Description
`0`	Data was transferred successfully
`1`	A local failure occurred on the client side and the tool did not transfer any files. The problem can be file permissions. Check the log file for more information, and check the file permissions before retrying the transfer.
`2`	The SAP Ariba data transfer tool was unable to connect to the SAP Ariba system and the tool did not transfer any files. This can be caused by a command issued on the server side. Either the connection is unavailable or there is another problem, such as the server cannot handle the request at the moment. Check the log file for more information before retrying the transfer.

Error Code	Description
3	The request from the SAP Ariba data transfer tool was rejected by the SAP Ariba system and the tool did not transfer any files. You might see this error code if you specified the wrong shared secret or an internal server failure occurred. Check the log file for more information, and make sure you correctly enter the shared secret.

Version information transfer

To assist with troubleshooting or working with SAP Ariba Customer Support personnel, the following environment information is sent with every SAP Ariba data transfer tool request to the SAP Ariba system:

- `ClientVersion`—the version of the SAP Ariba data transfer tool
- `ClientInfo`—a concatenation of the following system properties:
  - `IP address/hostname`—IP address and host name of the computer running the SAP Ariba data transfer tool
  - `os.name`—operating system of the server running the SAP Ariba data transfer tool
  - `os.version`—version of the operating system running the SAP Ariba data transfer tool
  - `os.arch`—architecture of the operating system running the SAP Ariba data transfer tool
  - `java.vm.version`—version of the virtual machine used to run the SAP Ariba data transfer tool

Running database operations from the SAP Ariba data transfer tool

Operations to be performed on the database can be controlled from the SAP Ariba data transfer tool. To run an event with an operation, you must create a new CSV file with the records to update. The name of this file must be the name of the file you want to update, with the operation (case sensitive) appended. For example, to run the Load operation on a file, the syntax is: `csvfilename_Load`. The operation is executed when you run the SAP Ariba data transfer tool.

With this functionality, you can run an event with two different operations. To do this, you only need one copy of the event, and you need to keep track of only one event name. The operation used by this event is controlled by the SAP Ariba integration toolkit, which makes it easier to track and control what the event is doing.

About deactivating records using the SAP Ariba data transfer tool

Earlier, to run an event with two different operations you needed to duplicate the events configuration and set the default operation on the new event to the second operation to be run. In addition, you needed to make the second event not UI visible to ensure that two versions of the event did not appear on the UI. You also needed to track which event used which operation, which can lead to errors.

The events associated with an operation are loaded in a map and the order of execution of events is not defined. To define an order of execution for events, you must add the events in the `MessageDependencies.table` file. This file contains information about the sequence of execution of events.

You can deactivate records in the database using the SAP Ariba data transfer tool. This allows you to implement incremental updates to master data, which is recommended practice.

Deactivating a record

Procedure

1. Specify the operation directly at the end of the file name that you want to update. For example, if you want to deactivate a record in the `Address` database, you create a new CSV file named `Address_Delete.csv` containing the record to be deactivated.
2. Run the SAP Ariba data transfer tool. This deactivates the record.

> **Note**
>
> You can also load some records at the same time as you deactivate other records. Add the records to be loaded to a separate CSV file named `Address_Load.csv`. Now, when you run the SAP Ariba data transfer tool, the records in this new file are loaded.

Using the DB Connector

In this section:

Process Overview

The DB Connector allows you to extract master data from your JDBC-based back-end system database and save it to CSV files. You can then use the SAP Ariba data transfer tool to import the data into your Ariba system. Conversely, you can use the DB Connector to extract transactional data from your Ariba system, save the data to CSV files, and then load the data into your back-end system database.

The DB Connector allows you to extract data from or load data into back-end systems that expose interface schemas. The DB Connector inserts data directly into the interface tables and retrieves data by querying either interface or production tables.

> **Note**
>
> The DB Connector is not designed for use with back-end systems, such as SAP, that do not expose interface schemas. For more information about how to integrate your SAP back-end system with your Ariba system, see the Ariba Procurement Solutions Integration Guide for SAP. The DB Connector is not a general mapping or integration tool. Mapping is limited to SQL expressions, and the tool does not allow integration through SAP iDocs, XML, or EDI. Only database table interfaces are supported.

Supported back-end system databases include:

- Oracle
- DB2
- SQL Server

In a typical installation, data transfers between your Ariba system and your back-end system occur at different intervals. For example, you might:

- Upload master data every day, or every other day
- Download payments every four to six hours

- Upload remittance data every four to six hours

Because of differences in frequency, direction, and contents of data transfers, you use a separate instance of the DB Connector to process each flow. For each flow, you create a properties file and a set of queries to execute.

Extracting Data from Your Back-End System Database

To extract data from your back-end system, you specify an output directory, database connection information, a log directory, and a query directory in the appropriate DB Connector properties file. For more information, see Creating Properties Files [page 77]. You also create an individual file for each query you want to run, which results in one generated CSV file per query.

On export, the DB Connector reads each query file in the query directory, executes the query, and generates a CSV file of the same name as the query file. Once all queries have finished executing, the DB Connector sends an email summarizing status and then terminates. If one or more queries fail, the DB Connector sends a status email summarizing the failure, but does not create any CSV files.

The following picture illustrates the process of extracting data from your back-end system using the DB Connector. The extracted data can subsequently be imported into your Ariba system using the SAP Ariba data transfer tool:

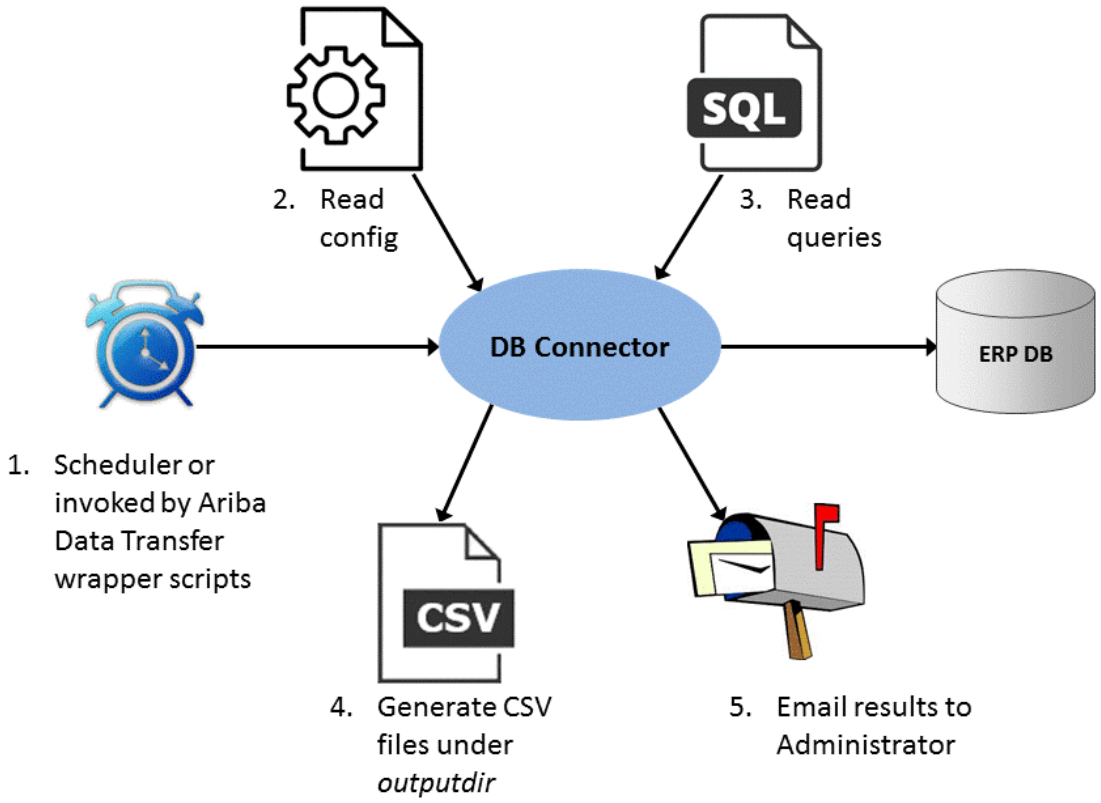


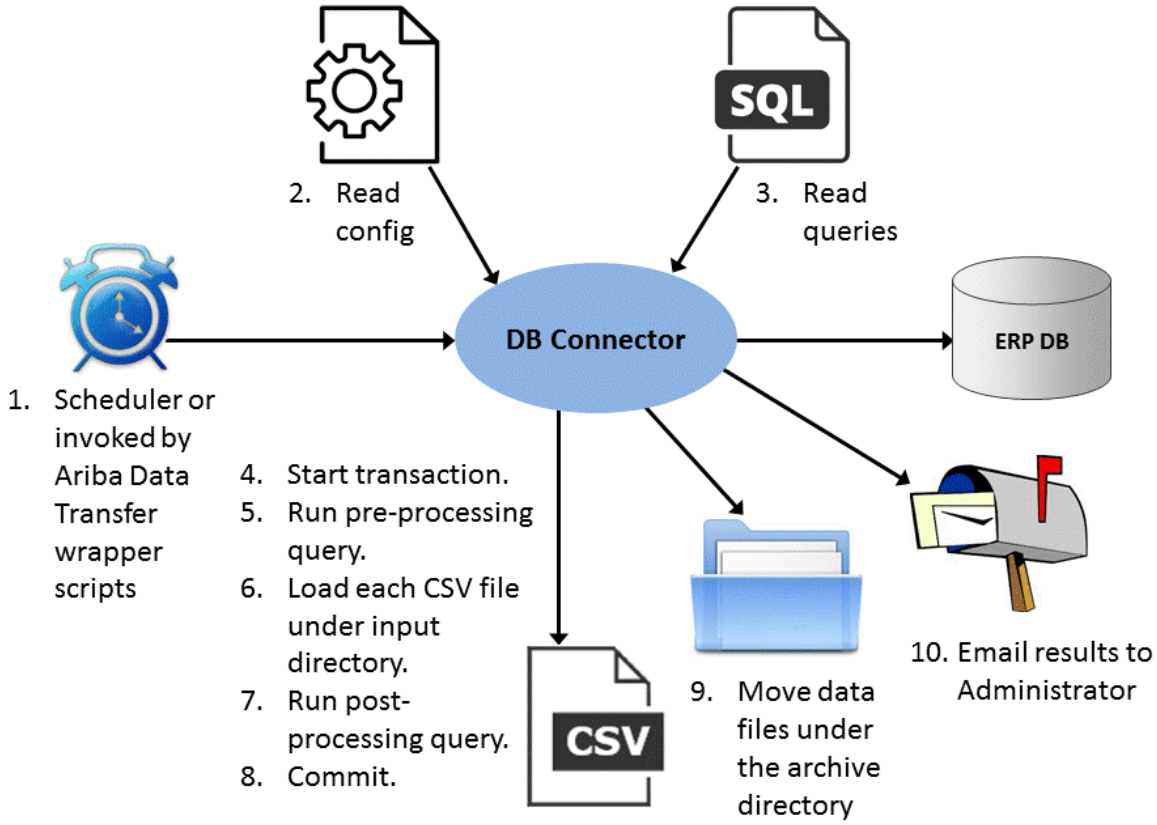
Figure 3: Extracting Data from Your Back-End System Using the DB Connector

Loading Data Into Your Back-End System

To load data into your back-end system, you specify an input directory, database information, a log directory, and a query directory in the appropriate properties file. You must also create an individual query file for each query you want to run.

Upon import, the DB Connector reads each query file and executes the query against the CSV file of the same name. Once all queries in the input directory have finished executing, the optional query contained in the `postprocessing.sql` file executes to process the loaded data set. Upon successful completion, the DB Connector commits the transaction, moves the data files into the archive directory, sends an email summarizing status, and then terminates.

The following picture illustrates the process of loading data extracted from your Ariba system into your back-end system using the DB Connector:



.

Figure 4: Loading Data into Your Back-End System Using the DB Connector

Related Information

Creating Properties Files [page 77]

Using the DB Connector

Prerequisites

These steps assume that you have already unpacked the tool. See Installing or upgrading the integration toolkit [page 10].

Procedure

1. Encrypt your database password for security. See About Encrypting The Database Password [page 74].
2. Establish directory structures to handle the various files upon which the tool relies. See Creating Directory Structures [page 76].
3. Create properties files for the each event group you plan to run. See Creating Properties Files [page 77].
4. Create the SQL queries that the DB Connector will run. See Creating SQL Query Files [page 81].
5. Run the DB Connector. The DB Connector can be launched from the command line or scheduled to run automatically. See Running the DB Connector [page 85].

Database Vendor Specific Implementation

If a database vendor has implemented a specific functionality, for example, enhanced security, then the DBConnector cannot exchange data with the database. In this case, ensure that the database client is installed on the machine where DBConnector is configured. The DBConnector can then use the database client to exchange data with the database.

For example, you are using the Oracle database that has a specific security implementation. In this case:

1. In the sqlnet.ora file of the Oracle client, the following entries may need to be configured:
```
SQLNET.ENCRYPTION_CLIENT=REQUIRED
SQLNET.ENCRYPTION_TYPES_CLIENT= (AES256)
SQLNET.CRYPTO_CHECKSUM_CLIENT=REQUIRED
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT= (MD5)
SQLNET.ENCRYPTION_SERVER = ACCEPTED
SQLNET.CRYPTO_CHECKSUM_SERVER = ACCEPTED
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER= (MD5)
SQLNET.ENCRYPTION_TYPES_SERVER= (AES256)
SQLNET.CRYPTO_SEED = 2z0hslkdharUJCFtkwbjOLbgwsj7vkqt3bGoUylihnvkhgkdsbdskkKGhdk
```
2. `databasehost` parameter in the oracle.property need to be set to
```
jdbc:oracle:oci:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=dbhostname)
(PORT=dbport))(CONNECT_DATA=(SERVICE_NAME=servicename))
```

About Encrypting The Database Password

For security purposes, you can encrypt the database password that you use in the properties files. The SAP Ariba integration toolkit comes with a utility script for this purpose called `encryptstring`. This script creates an encrypted version of the input string, which it displays at the command prompt so that you can copy the encrypted string into all the properties files that require it.

When you are developing and testing your DB Connector setup, you might prefer to work with an unencrypted database password for ease of use. However, Ariba recommends that all database passwords be encrypted in actual production situations for security.

The location of `java.exe` is required for the `encryptstring` script.

Encrypting the Database Password

Procedure

1. Install Java Runtime Environment 1.6, if it is not already installed.
2. Create the options file using the appropriate steps for your operating system.
   - **For Microsoft Windows:**
     1. Using Notepad or a similar text editor, create a new file called `dbconnector_env.bat`.
     2. Set the `JAVA_HOME` option to the path up to, but not including the bin directory. For example: set JAVA_HOME=C:\Program Files\Java\<jre version>
     3. Save and close the file.
   - **For UNIX:**
     1. Using VI or a similar text editing tool, create a new file called `dbconnector_env.sh`.
     2. Write a line setting the environment variable `JAVA_HOME` to the Java installation directory. For example:
        `export JAVA_HOME=/usr/local/`<jre version>
     3. Save and close the file.
3. At the command prompt change to the directory containing the `encryptcustomerkey` script. For example:

   ```
   cd D:\IntegrationToolkit\bin
   ```

4. Run the script to encrypt the customer key using the following syntax:

   ```
   encryptcustomerkey.bat <options file>
   ```

   where `<options file>` is the path and name of an options file where you have set the `JAVA_HOME` option.

   For example: `D:\IntegrationToolkit\bin>encryptcustomerkey.bat "D:\IntegrationToolkit\sample-options\dbconnector_env.bat"`

   A prompt to enter the key is displayed.
5. Enter the customer encryption key to be encrypted.

The key must be an alphanumeric string of 24 characters in length. For example, `TestCustomerKey123456789`.

The encrypted key is displayed as an output on the command line. For example, `{3DES}+Y9xa4mwCKKvBh6oPRIFvkdk2srHINb/I0JygsfYQWQ=`.

> **Note**
>
> The encrypted string in this example is prefixed with the string `{3DES}`, indicating the encryption algorithm used is Triple DES. The DB Connector uses the entire encrypted string.

6. Copy the encrypted key from the command line to a.txt file and save the file. For example, `CustomerKey.txt.`

   The encrypted key must be the first and only line in the file without any spaces before or after. The path to this file must be passed as the value for the `encryptionKeyPath` parameter in the options file.

7. Run the encryptstring script using the following syntax:

   ```
   bin\encryptstring.bat dbconnector_env.bat
   ```

   A prompt to enter the password is displayed.

8. Enter the database password to be encrypted.

   The encrypted password is displayed as an output on the command line. For example, `{3DES}+Y9xa4mwCKKvBh6oPRIFvkdk2srHINb/I0JygsfYQWQ=`.

9. Copy the encrypted password from the command line to a.txt file and save the file. For example, `DatabasePassword.txt.`

   The encrypted database password must be the first and only line in the file without any spaces before or after. The path to this file must be passed as the value for the `databasePasswordFilePath` parameter in the properties file.

10. In the DB Connector properties file, pass the path of the `CustomerKey.txt` file for the `encryptionKeyFilePath` parameter as follows:

    ```
    <set encryptionKeyFilePath=D:\IntegrationToolkit\CustomerKey.txt>
    ```

11. In the DB Connector properties file, pass the path of the `DatabasePassword.txt` file for the `databasePasswordFilePath` parameter as follows:

    ```
    <set databasePasswordFilePath=D:\IntegrationToolkit\DatabasePassword.txt>
    ```

12. Save the file.

Securing Your Database

In addition to password encryption, there are other security measures that you must implement to secure your database when using the DB Connector. These measures include:

- Carefully managing directory permissions
- Carefully managing file permissions
- Using a secure database account

The properties files used by the DB Connector contain database passwords. These passwords, even though encrypted, must be protected from view by unauthorized users. All files must be visible only to the administrative user running the DB Connector, and must not be visible to other users or groups.

Properties files must have only read and write permissions for the account running the DB Connector, but no permissions for any other accounts.

The database account used to run the DB Connector must be a restricted user that deals only with integration. Allowing limited access helps prevent unintended changes to the data in your back-end system.

Likewise, data and query files contain sensitive business data that must not be exposed to unauthorized users. Ensure that only the software consuming the files has access to them.

Creating Directory Structures

Before running the DB Connector, you must first create directory structures to handle the various files based on which the tool runs. When you create your properties files as described in Creating Properties Files [page 77], you specify the locations of the directories you create. These directories include:

- `config`, where the properties file is located.
- `querydir`, where the SQL queries for that event group are located.
- `outputdir`, for data export. Indicates where the CSV files generated by the SQL queries are located. The DB Connector creates this directory automatically.
- `inputdir`, for data to be imported into your back-end system. Indicates where the CSV files exported from your Ariba system are stored.
- `logdir`, where log files are stored.
- `archivedir`, where archives are stored. `The` DB Connector creates this directory automatically.

The proper directory structure:

- Allows the isolation of queries and data files from event to event
- Provides all information in a single location for the administrator to configure
- Is compatible with the SAP Ariba data transfer tool

> **Note**
>
> You must create a separate directory structure for each site in which the SAP Ariba integration toolkit operates, using the architecture displayed in the following example as a guide.

The following example explains how to configure the directories for a scenario in which you extract master data from the back-end system database and upload transactional data (in this case, payments) to the back-end system database. In this example, the DB Connector is installed in the following directory:

```
d:/IntegrationToolkit
```

The directory structure for the scenario is:

```
d:/IntegrationToolkit
    Config
    OracleFinancials
        MasterData
```

```
            Queries  --> querydir
            Pending --> outputdir
            Logs      --> logfile
        Payments
            Queries  --> querydir
            Pending  --> inputdir
            Archives --> archivedir
            Logs --> logFile directory
```

In the master data properties file, the following properties are specified:

```
querydir=d:/IntegrationToolkit/OracleFinancials/Masterdata/Queries
outputdirdir=d:/IntegrationToolkit/OracleFinancials/Masterdata/Pending
LogFile=d:/IntegrationToolkit/OracleFinancials/Masterdata/Logs/md.log
```

In the transactional data properties file, the following properties are specified:

```
querydir=d:/IntegrationToolkit/OracleFinancials/Payments/Queries
outputdir=d:/IntegrationToolkit/OracleFinancials/Payments/Pending
ArchiveDir=d:/IntegrationToolkit/OracleFinancials/Payments/Archives
LogFile=d:/IntegrationToolkit/OracleFinancials/Payments/Logs/md.log
```

Timestamp Conventions

The `outputdir` data directory produced by the DB Connector follows this naming convention:

```
dbconnector<timestamp1>
```

Once .zip files are moved by the SAP Ariba data transfer tool, the directory name becomes:

```
dbconnector<timestamp1>.<timestamp2>
```

Related Information

Creating Properties Files [page 77]
Incremental Queries [page 84]

Creating Properties Files

You configure the DB Connector by creating `.properties` files for each event group extract or load you plan to run. All sample properties files for the DB Connector are stored in the `DBConnector` directory which was created when you unpacked the SAP Ariba integration toolkit. Which sample files you use depends upon which database you are using, and which event group you require.

For each file you create, you must save the file and store it in the `config` directory. For example:

```
cd d:/IntegrationToolkit
```

```
mkdir config
cd config
cp ../dbconnector.properties.sample dbc-masterdata.properties
```

Next, you edit the properties file as described later in this section. When the DB Connector is run, the tool retrieves the information it requires to execute the transaction.

Related Information

Creating Directory Structures [page 76]

Sample Properties Files

To specify properties for exporting transactional data from your Ariba system, use one of these sample files:

- `DBConnector\data-export\db2.properties`
- `DBConnector\data-export\oracle.properties`

To specify properties for importing master data, incremental master data, and transactional data into your Ariba system, use one of these files:

- `DBConnector\data-import\db2.properties`
- `DBConnector\data-import\oracle.properties`

Specifying Properties

The following tables describe the options in the `.properties` files. Note that the system supports only forward slashes ('/') in directory paths.

Property	Description
`querydir`	The directory where the SQL files are located. For example: `querydir=D:`/IntegrationToolkit`/dbc/db2_psoft/md/queries` or querydir=D:/IntegrationToolkit/dbc/ora_sg/md/queries **Note** When pushing data to your Oracle ERP system, you need to create a sequence file in the query input directory. In the sequence file, enter the SQL query filenames separated by a semicolon (;) based on the sequence in which the queries need to be executed.

Property	Description
`outputdir`	The location of the CSV files created to hold the data extracted from the back-end system. For example: `outputdir=D:/IntegrationToolkit/dbc/db2_psoft/md/datafiles` or outputdir=D:/IntegrationToolkit/dbc/ora_sg/md/datafiles/output This option is not required for data import.
`inputdir`	The location where the CSV files to be imported into the back-end system are stored. This option is not required for data export.
`archivedir`	The location to which completed transaction directories and files must be moved. For example: `archivedir=D:/IntegrationToolkit/dbc/db2_psoft/md/finished` or archivedir=D:/IntegrationToolkit/dbc/ora_sg/md/finished
`databasedriver`	The database driver. For example: databasedriver=com.ibm.db2.jcc.DB2Driver or databasedriver=oracle.jdbc.driver.OracleDriver
`databasehost`	The JDBC information needed to connect to the database. The syntax for this option will vary for each database type. Please ensure that the value for this option matches the required syntax for your database. For example: `databasehost=jdbc:db2://zodiac.yourcompany.com:50020/ad820dev` or databasehost=jdbc:oracle:thin:@172.29.129.55:1521:VIS
`databaseusername`	The login ID for the database. For example: `databaseusername=jdoe`
`databasepassword`	The encrypted database password. For example: `databasepassword={3DES}oHX+gzt9Z40=`
`databasemapping`	The path to the file containing query syntax for the database. Syntax will vary for each database, so each database type must have its own file. For example: `databasemapping=D:/IntegrationToolkit/dbc/dbctool/DBSyntax.table.db2.properties` or databasemapping=D:/IntegrationToolkit/dbc/dbctool/DBSyntax.table.oracle.properties
`csvencoding`	The type of CSV encoding to be used. For example: `csvencoding=Cp1252`

Property	Description

`logFile`	The location of the log file. For example:
	`logFile=D:/IntegrationToolkit/dbc/db2_psoft/db2-sg-md.log`
	or
	`logFile=D:/IntegrationToolkit/dbc/ora_sg/ora-sg-md.log`

`onemptylines`	Specifies how to deal with error reporting of empty lines in CSV files. Possible values include `SkipLine`, `WarnUser`, and `TreatAsException`. For example: `onemptylines=SkipLine`
`onemptyfile`	Specifies how to deal with error reporting of empty CSV files. Possible values include `WarnUser`, and `AbortPull`. For example: onemptyfile=WarnUser
`oninvaliddata`	Specifies how to deal with error reporting of invalid data. Possible values include `SkipLine`, `WarnUser`, and `TreatAsException`. For example: `oninvaliddata=TreatAsException`
`smtp`	The name of the SMTP mail server to be used for email notifications. For example: smtp=smtp.domain.com
`to`	The email address to be used for notifications. For example: to=user@domain.com
`from`	The email address from which notifications are to be sent. For example: from=db-connector@domain.com
`timestampquery`	The query used to retrieve the current time from the database. For example: timestampquery=values(current timestamp) or timestampquery=SELECT to_char(SYSDATE, 'DD-Mon-YYYY HH24:MI:SS') as \"time\" from dual
`timestampformat`	The format of the time returned by the query. This format must follow the SimpleDateFormat as specified in the Java API. For example: timestampformat=yyyy-MM-dd HH:mm:ss
`fromDate`	The string to find in the query for the "from" timestamp. For example: fromDate=_from This option is optional. If not specified, use `!fromDate!` notation.
`toDate`	The string to find in the query for the "to" timestamp. For example: toDate=_to The option is optional. If not specified, use `!toDate!` notation.

Related Information

About Encrypting The Database Password [page 74]
Creating SQL Query Files [page 81]

Configuring Email Notification

You can configure the DB Connector to provide email notification for both successful and failed runs. To enable email notification, uncomment and configure the three email notification parameters in your `<database>.properties` file(s):

- `smtp`: The name of the SMTP mail server to be used for email notifications. For example:

  ```
  smtp=smtp.domain.com
  ```

- `to`: The email address to be used for notifications. For example:

  ```
  to=user@domain.com
  ```

- `from`: The email address from which notifications are to be sent. For example:

  ```
  from=db-connector@domain.com
  ```

Creating SQL Query Files

For each event group that you plan to run, you create individual `.sql` files that are stored in the `querydir` directory you created earlier. These queries can be structured in a database-specific way, or you can write abstracted queries using the sample database syntax mapping files provided by Ariba.

The query files you write must be stored in the `querydir` directory so that the DB Connector can find and execute them.

Each file must contain one query formatted on a single line with no semicolon at the end of the line.

Sample Queries

The following example shows a query that extracts suppliers from an Oracle system.

> **Note**
>
> This query is formatted for documentation purposes. In the actual query file, the query appears on a single line:

```
Select vend.vendor_id as "UniqueName",
    vend.vendor_name as "Name" From po_vendors vend
```

```
Where
    (not vend.vendor_type_lookup_code = 'EMPLOYEE')
    AND ((vend.start_date_active < sysdate) or (vend.start_date_active is null))
    AND ((vend.end_date_active >= sysdate) or (vend.end_date_active is null))
    AND (vend.employee_id is null)
    AND vend.enabled_flag = 'Y'
    AND vend.vendor_id in
        (select site.vendor_id from po_vendor_sites_all site, po_vendor_contacts
                cont,org_organization_definitions org
        where (vend.vendor_id = site.vendor_id)
            AND (cont.vendor_site_id = site.vendor_site_id)
            AND (site.purchasing_site_flag = 'Y')
            AND ((site.inactive_date >= sysdate) or (site.inactive_date is null))
            AND ((cont.inactive_date >= sysdate) or (cont.inactive_date is null))
            AND (site.org_id = org.operating_unit)
            AND (org.chart_of_accounts_id = @COAID@))
```

Notice the following:

- When extracting data from your back-end system database and generating CSV files, the query must specify the CSV file column in which the values from each column in the database table are stored. You accomplish this specification using column aliases in the query, where the name of the alias is the name of the column header in the CSV file that the DB Connector generates. In the example above, `UniqueName` is the column name in the CSV file for the back-end system column `vendor_id`.
- In the last line of the example above, `@COID@` is a parameter that is replaced during the execution of the query by a value configured in the database-specific syntax mapping file.

The following query example inserts payment headers into Oracle. Again, note that in the actual query file you create, the query appears on a single line.

```
INSERT INTO ap_invoices_interface
            (invoice_id, invoice_num, invoice_type_lookup_code,
            invoice_date, vendor_id, vendor_site_id, invoice_amount,
            invoice_currency_code, terms_id, source, org_id,
            attribute1, payment_method_lookup_code)
        VALUES (ap_invoices_interface_s.NEXTVAL,
                    !InvoiceNumber!,'STANDARD',
                    TO_TIMESTAMP_TZ(!InvoiceDate!,'mm/dd/yyyy HH24:MI:SS tzh:tzm'),
                    !Supplier!,
                    !SupplierLocation!,
                    !GrossAmount!,
                    !GrossAmountCurrency!,
                    !PaymentTerms!,
                    'ARIBA',
                    (substr(!PurchasingUnit!,1,instr(!PurchasingUnit!,':')-1)),
                    !InvoiceReconciliationId!, upper(!PaymentMethod!))
```

When loading the data from a CSV file, you must specify exactly which column from the CSV file will be inserted into which column in the database table. You make this specification using the special exclamation mark delimited notation in the query. In the example above, the values in the `InvoiceNumber` column in the CSV file, referred to as `!InvoiceNumber!` in the query, are inserted into the `invoice_num` column in the back-end system database table.

Related Information

Mapping SQL Syntax [page 83]

Mapping SQL Syntax

Ariba provides two SQL syntax mapping files that abstract RDBMS-specific SQL syntax by providing string substitutions for common differences between SQL versions. These files provide default configurations for the three supported databases: Oracle, DB2, and SQL Server.

You can also extend the mapping syntax files according to your needs. For example, you can parameterize queries to provide either a range of dates for data pull, or provide account layout parameters for Oracle.

Use the appropriate file for your database, either:

- `DBConnector\sql-syntax-mapping\DBSyntax.table.mssql.properties`
- `DBConnector\sql-syntax-mapping\DBSyntax.table.oracle.properties`

To use abstraction parameters in your query files, enclose the parameters between "@" signs.

For example, the following Oracle-specific query:

```
Select sysdate from dual
```

can be abstracted as:

```
Select @CurrentDate@ from MyTable
```

Alternatively, you can extend the `DBSyntax.table.oracle.properties` file with the following parameter:

```
TimeQuery = "Select sysdate from dual"
```

and then write your query file like this:

```
@TimeQuery@
```

Depending on how you use the syntax mapping file, that is, for SQL syntax abstraction or to specify query parameters, you can store the file in the `config` directory or in the `query` directory itself. Regardless of where you store it, the `databasemapping` parameter in the properties file must point to the fully qualified path of the syntax mapping file.

Mapping Within Oracle Character Limits

This section pertains to Oracle users only.

In the query examples shown earlier in this chapter, column aliases were used to map the Oracle table names to the CSV column names. Oracle truncates the alias name to 30 characters if you specify anything over that length. To work around this limitation, add a property file for each query. Each property file contains the mapping between the Oracle column name and the CSV column names.

For example, a query called `CurrentDate.query` that returns a CSV column name exceeding the Oracle limit can be abstracted in a file called `CurrentDate.properties` as:

```
CURRENTDATE=MyVeryLongCSVColumnName
```

Note that the database column name in the mapping file is specified in capital letters to match the column name shown in the database result set.

Incremental Queries

The DB Connector pulls all master data and loads it into your Ariba system. In the case of transactional data, the DB Connector pulls only the delta from the last data pull. This process prevents the sending of duplicate information to your Ariba system. To establish when the last pull occurred, the DB Connector stores a timestamp file each time it runs a query. This file, `timestamp.txt`, is saved in each query directory, and contains the full timestamp (Pacific Standard Time) of the last query execution.

Incremental extract is always enabled. A 30 minute offset is applied to the database time to avoid pulling incomplete or inconsistent data.

Any query can become incremental by adding a `where` clause using the tags `!fromDate!` and `!toDate!`. Note that incremental queries with arbitrary dates specified on the command line are not supported.

Creating a Baseline Timestamp

A timestamp is written in each query directory on the first run of the query. During the initial extract, no data is loaded unless a start date is provided. To create a baseline timestamp, you create and run a query that returns only the system date. For example:

```
Select sysdate from dual
```

Next, modify the query to include today's date. For example:

```
Select sysdate as CurrentDate from dual
```

Run the query again and check to see that the file `outputdir/CurrentDate.csv` is created.

Pre- and Post-processing Queries

Depending upon your situation, the data you extract might require pre- or post-processing. For example, you might need to join the header and detail records with back-end system-specific join keys after loading data.

You can create pre- and post-processing SQL files that contain several statements to run on the data set. In the case of post-processing, these queries run after all the CSV loading queries have completed.

Pre- and post-processing queries must be contained in files called `preprocessing.sql` and `postprocessing.sql`. These files must be located in the `query` directory, along with all the other queries for a particular event group.

Running the DB Connector

The command-line syntax used to run the DB Connector varies according to how you use the tool. Sample scripts are provided in:

```
<installation_directory>/etc/sample-bin
```

To run the DB Connector, you must change the variables in the script to point to the appropriate locations, save the script, and then run the script.

Error Handling

The DB Connector returns the following codes to indicate the success or failure of a particular query:

Error Code	Description
0	Success
1	A local failure occurred on the client side. The problem might be file permissions. Check the log file for more information, and check the file permissions before retrying.
2	A remote temporary failure occurred. This can be caused by a command issued on the server side. Either the connection is unavailable or there is another problem, such as the server cannot handle the request at the moment. Check the log file for more information before retrying.
3	A remote permanent failure occurred. You might see this error code if you specified the database password or an internal server failure occurred. Check the log file for more information.

Data is always made available automatically, that is, all SQL files generated successfully are made available. In the case of a rollback, no files are made available.

If you have configured email notification, an email is sent containing an explicit error message, resolution notes, and the location of the relevant properties files and log files.

Related Information

Configuring Email Notification [page 81]

SAP Ariba integration toolkit on Oracle Fusion Middleware

In this section:

About the SAP Ariba integration toolkit on Oracle Fusion Middleware

This is available for SAP Ariba Invoice Management.

.

The Ariba Network Adapter for Oracle Fusion Middleware allows you to exchange transactional data between Ariba Network and your Oracle Financials ERP system. If you are using SAP Ariba Invoice Management for invoice automation, then invoices from Ariba Network are sent to SAP Ariba Invoice Management. To ensure that invoices from Ariba Network are reconciled in SAP Ariba Invoice Management, you need to download master data corresponding to these invoices from your Oracle Financials system.

If you use the SAP Ariba integration toolkit to download master data, then you are required to maintain and monitor two different environments, one for master data and the other for transactional data integration. This is cumbersome and time-consuming. Therefore, instead of using the SAP Ariba integration toolkit as a standalone tool, you can now deploy the SAP Ariba integration toolkit on Oracle Fusion Middleware.

This provides the following benefits:

- You have a single solution with Oracle Fusion Middleware for both master and transactional data integration.
- You can leverage Oracle Fusion Middleware's monitoring, transaction tracking, and error handling capabilities.

The SAP Ariba integration toolkit is executed through the Quartz Scheduler and the inputs for execution of jobs are provided using XML files. Quartz is an open source scheduler (http: www.quartz-scheduler.org) and the version used is Quartz 1.7.2.

> **Note**
>
> The SAP Ariba integration toolkit on Oracle Fusion Middleware consists of the SAP Ariba data transfer tool only. The DB Connector is not part of the implementation.

Enabling the SAP Ariba integration toolkit on Oracle Fusion Middleware

To enable the SAP Ariba integration toolkit to be executed from Oracle Fusion Middleware, you must deploy the .EAR file to Oracle Fusion Middleware. See Managing Ariba Integration Toolkit on Oracle Fusion Middleware [page 87].

> **Note**
>
> Currently, the EAR file can be deployed on Oracle SOA Suite version 10.1.3.4. It can also be deployed on a standalone Oracle Application Server version 10.1.3.1.

Using the SAP Ariba integration toolkit on Oracle Fusion Middleware

SAP Ariba provides a ZIP file (`aribaoc4jfiletransfer.zip`) that contains an EAR file and six sample template XML files that correspond to the six job definitions as follows:

- Full Download
- Full Upload
- Incremental Download
- Incremental Upload
- ANUpload
- ANDownload

The first four job definitions are applicable to SAP Ariba Invoice Management and SAP Ariba Sourcing. The remaining two jobs definitions are relevant for Payment Integration with Ariba Network.

The ZIP file contains another ZIP file (aribaitklib.zip) that contains all the third-party JAR files that are required to deploy the `aribaoc4jfiletransfer.zip` on Oracle Fusion Middleware.

You can configure the execution of jobs through the XML files by passing values to the parameters that are available in these XML files.

Managing the SAP Ariba integration toolkit on Oracle Fusion Middleware

In this section:

Using the SAP Ariba integration toolkit on Oracle Application Server

Procedure

1. Download the `aribaoc4jfiletransfer.zip` file from connect.ariba.com.
2. Login to connect.ariba.com and click On-Demand > SAP Ariba Invoice Management > Ariba Integration Toolkit for Oracle Fusion Middleware.
3. Save the `aribaoc4jfiletransfer.zip` on your file system.
4. Unzip the deployable `aribaoc4jfiletransfer.zip`.

   The Bin folder contains the `encryptcustomerkey` and `encryptstring .bat` and `.sh` files. Based on your operating system, you can encrypt your authentication credentials.
5. Deploy the third-party JAR files on the OC4J container.
6. Configure the relevant job XML files.
7. Deploy the EAR file.

Related Information

Encrypting Your Passwords [page 131]

Deploying the Third-Party JAR files on Oracle Application Server

Procedure

1. Extract the `aribaitklib.zip` provided in `aribaoc4jfiletransfer.zip`.
2. Log on to Application Server Control.
3. Click the oc4J (Oracle Container for J2EE) instance on which you have deployed the adapter.
4. Click the **Administration** tab.

5. Create a shared library called `ariba.itk.lib` and specify version as `1.0`.

   > **Note**
   >
   > The name of the shared library must be `ariba.itk.lib`.

6. Click on go to task for **Shared Libraries** under **Administration Tasks - properties**
7. Click on **Create** button.
8. Enter `ariba.itk.lib` as Shared Library Name and `1.0` for Shared Library Version
9. Click **Next** and click **Add**.
10. Select `File is present on local host. Upload file to this shared library's directory on the target OC4J instance.` and click on **Browse** button to select the jar file from the folder in which `aribaitklib.zip` file was extracted.
11. Click **Continue**.
12. Repeat steps 10 and 11 until all the third-party libraries are added and then click **Finish**.

Configuring the XML file for a Job on Oracle Application Server

Procedure

1. Open the XML file corresponding to the job you want to configure.
2. In the **Job detail** section, specify values for the following elements:

   <name> - Name of the job

   <group> - Group to which the job belongs. For example, you can have two groups, Ariba_Upload and Ariba_Download, for upload and download jobs respectively.

   <description> - A short description of the job
3. Enter values for all relevant parameters.

   > **Note**
   >
   > To encrypt the shared secret, you must use the encryptstring.bat or sh from the command line. The encrypted shared secret and key can then be used as parameter values.

4. In the **Trigger** section, specify values for the following elements:

   <name> - Name of the trigger

   <group> - Group to which the trigger belongs

   <job-name> - Name of the job

   <job-group> - Group to which the job belongs

   <cron expression> - The schedule of the job

   For information on the different cron expressions that you can specify, see http://www.quartz-scheduler.org/docs/tutorials/crontrigger.html

5. Save the file.

> **Note**
>
> If you are configuring multiple uploads or multiple downloads, then you must provide a unique job name and cron job name for each upload or download transaction.

Related Information

Scheduled Job Parameters [page 93]

Deploying the EAR File on Oracle Application Server

Procedure

1. Log on to Application Server Control.
2. Click the oc4J (Oracle Container for J2EE) instance on which you have deployed the adapter.
3. Click on the **Application** tab.
4. Click on **Deploy**.
5. On the **Wizard Screen Deploy: Select Archive**, provide the path of the EAR file (`aribaoc4jfiletransfer.ear`). This EAR file is available in the `aribaoc4jfiletransfer.zip` that you downloaded earlier.
6. Click **Next**.
7. On the **Wizard Screen Deploy: Application Attributes**, provide a value for the **Application Name**.
8. Click **Next**.
9. On the **Wizard Screen Deploy: Deployment Setting**, click on **Edit Deployment Plan** and from the left side navigation, navigate to **Oracle Enterprise Manager** -> **War** -> **ITK Scheduler** -> **context param** -> **jobfilepath**. Enter the comma separated file paths of the XML files that you configured earlier.

> **Note**
>
> If you want to execute only one job, enter the path of the that file only.

10. Click **OK**.

> **Note**
>
> There is one more parameter called quartzProperties that allows an advanced user more granular control on the Quartz Scheduler. To use this parameter, the path of the Quartz.Properties file can be passed in step 9 along with the path of the XML files.

11. Click **Deploy**.

> **Note**
>
> You can edit the job XML even after deployment as the Quartz Scheduler is scheduled to read the XML files every two minutes.

Using the SAP Ariba integration toolkit on Oracle WebLogic

Procedure

1. Download the `aribawlfiletransfer.zip` file from connect.ariba.com.
2. Login to connect.ariba.com and click On-Demand > SAP Ariba Invoice Management > Ariba Integration Toolkit for Oracle Fusion Middleware.
3. Save the `aribawlfiletransfer.zip` on your file system.
4. Unzip the deployable `aribawlfiletransfer.zip`.

   The Bin folder contains the `encryptcustomerkey` and `encryptstring .bat` and `.sh` files. Based on your operating system, you can encrypt your authentication credentials.
5. Generate a deployment plan.
6. Configure the relevant job XML files based on the jobs you want to execute.
7. Deploy the Enterprise Archive (.EAR) file.

Related Information

Encrypting Your Passwords [page 131]

Generating a Deployment Plan on Oracle WebLogic

Procedure

1. Run the following script from the command line for Windows:

```
set BEA_HOME=<Middleware home>
call %BEA_HOME%\wlserver_10.3\server\bin\setWLSEnv.cmd
set PATH=%BEA_HOME%\<jre>\bin
set planname=itkplan.xml
set earfile=<path to ear>\aribawlfiletransfer.ear
java weblogic.PlanGenerator -all -plan %planname% %earfile%
```

2. Run the following script for Linux:

```
export BEA_HOME=<Middleware home>
$BEA_HOME\wlserver_10.3\server\bin\setWLSEnv.sh
```

```
export PATH=$BEA_HOME\<jdk>\bin
export planname=itkplan.xml
export earfile=<path to ear>\aribawlfiletransfer.ear
java weblogic.PlanGenerator -all -plan $planname $earfile
```

This generates a deployment plan called `itkplan.xml`.

3. Open this `itkplan.xml` file, under `<variable-definition>`, add the following tags:

```
<variable>
<name>jobFilePath</name>
<value>Path To the JOB.XML file</value>
</variable>
```

> **Note**
>
> In the above tag, replace the text "Path to the job.xml" with the XML input job files that you created. In case of more than one job file, put comma separated values.

4. Search for the element, `<module-override> <module-`.
5. Under this element, the following tag is available:

```
<root-element>web-app</root-element> <uri>WEB-INF/web.xml</uri>
```

6. 5 Paste the following tags, after the tag mentioned in the previous step:

```
<variable-assignment> <name>jobFilePath</name>
<xpath>/web-app/context-param/[param-name="jobFilePath"]/param-value</xpath>
<operation>replace</operation> </variable-assignment>
```

7. Save the file.

Related Information

Using the SAP Ariba integration toolkit on Oracle Application Server [page 88]

Deploying the .EAR File on Oracle WebLogic

Procedure

1. Run the following script from the command line for Windows:

```
set BEA_HOME=<Middleware home>
call %BEA_HOME%\wlserver_10.3\server\bin\setWLSEnv.cmd
set PATH=%BEA_HOME%\<jre>\bin
set planname=itkplan.xml
set earfile=<path to ear>\aribawlfiletransfer.ear
java weblogic.Deployer -adminurl t3://<hostname:port> -user <username> -password
<password> -deploy -name AribaFileTransfer -source %earfile% -targets <weblogic
server> -stage -plan %planname%
```

2. Run the following script for Linux:

```
export BEA_HOME=<Middleware home>
$BEA_HOME\wlserver_10.3\server\bin\setWLSEnv.sh
export PATH=$BEA_HOME\<jdk>\bin
export planname=itkplan.xml
export earfile=<path to ear>\aribawlfiletransfer.ear
export CLASSPATH=<path to the weblogic.jar in the machine under BEA HOME>
java weblogic.Deployer -adminurl t3://<hostname:port> -user <username> -password
<password> -deploy -name AribaFileTransfer -source $earfile -targets
<weblogicserver> -stage -plan $planname
```

Scheduled Job Parameters

Option	Description	Used For

`urlPrefix`	Examples values are:
	Ariba Procurement solution - `https://s1.ariba.com/Buyer`
	Ariba Sourcing - `https://s1.ariba.com/Sourcing`
	Ariba Network - https://service.ariba.com/ClientToolIntegrator.aw/ad/clienttool
	Using Digital Certificate-Based authentication, prefix the URL with the word `cert` as follows:
	`https://certs1.ariba.com/Buyer`
	`https://certs1.ariba.com/Sourcing`
		Upload, Download

`site`	The name of your company as used in the URL for your Ariba product. The URL is in the for `http://` **`mycompany`**`.procurement.ariba.com.` Example: `set site=mycompany`	Upload, Download

Option	Description	Used For
`serviceMode`	Specifies what kind of Ariba product you are using. Possible values include: • `0` to specify an Ariba On-Demand solution. The `urlPrefix` option will, in this case, refer to a site. For example: `http://s1.ariba.com/Buyer/fileupload?realm=realm_1` • `-1` is the default value, which must be changed to the appropriate allowed value to avoid a warning message. The warning message will be issued to the client log, followed by a permanent failure 4xx error code. Site and Service Mode are parameters relevant for jobs related to Ariba procurement solutions and Ariba Sourcing. ○ AribaIntegrationEventFullDownload ○ AribaIntegrationEventFullUpload ○ AribaIntegrationEventIncrementalDownload ○ AribaIntegrationEventIncrementalUpload For AribaANUpload and AribaANDownload jobs, there are three additional parameters: • Community - The community affiliation of the buyer. • System - The ERP ID configured on Ariba Network for multi-ERP support. • ANID - Specify the ANID of the buying organization. If the buying organization has more than one ERP system, define an application system ID to uniquely identify the ERP system in Ariba SN. For example: set ANID=AN0100000001 set systemID=ERP020	Upload, Download
`encryptionKeyPath`	Specifies the location of the file that contains the encrypted customer key.	Upload, Download
`sharedsecret`	Specifies the shared secret that is configured under Integration Toolkit Security in Ariba Administrator.	Upload, Download
`event`	Specifies an integration task. To use an integration task with the SAP Ariba data transfer tool, you must configure the task on the server. If you are using Microsoft Windows, you must quote the value of the `event` option. For more information about data import and export tasks, see the import and administration guide for your Ariba solution. For information about the form and contents of CSV files, see Data Dictionary.	Upload (transactional only), Download
`adaptersource`	The value of adapter source can be ALL, EXTERNAL, SYSTEM, or ARIBA MANAGED.	Download

Option	Description	Used For
`downloaddir`	Use to specify the directory (named `DownloadFromAriba` by default) containing the downloaded files in directories to be stored for processing by your back-end system. On Microsoft Windows, the directory is created in `C:\TEMP`. On UNIX, it is created in `/var/tmp`.	Download
`timestampFile`	For incremental downloads, specifies the time stamp file that ensures that the download event pulls data saved since the latest successful download.	Download
`filePrefix`	Use to specify a prefix for downloads, for example `expense`, `payment`, or `accruals`. The tool appends a time stamp to downloaded files, indicating when the download was completed.	Download
`notifyemail`	Specify a comma-delimited list of email addresses to which notifications must be sent, if you have enabled notifications for successful or failed data transfers. Example: `set notifyemail=myemail@mycompany.com`	Upload, Download
`smtpMailServer`	Specify the Simple Mail Transfer Protocol (SMTP) server that the Ariba Data Transfer Tool email client uses to send notification messages. Example: `set smtpMailServer=mail.mycompany.com`	Upload, Download
`topDir`	Specify the full path to the directory where the various input and output subdirectories must go. Example: `set topDir=C:\TEMP\FullMasterUpload`	Upload
`inDir`	The folder in which data files are placed for upload to Ariba. Specify the **full path** to the subdirectory containing the CSV files to be uploaded. The tool expects to find a subdirectory containing the CSV files to be transferred. Example: `set inDir=%topDir%\InDir\`*`your_subdirectory`*	Upload
`filterprefix`	Specify the subdirectory name prefix that indicates which CSV file subdirectories (in `inDir`) the tool must ignore when uploading files. Default: `set filterprefix=processing` If you have a program that generates CSV files, it must use a file name that begins with this prefix and then rename the file when the file is ready for transfer. If this task is scheduled to run while the file is still being generated, the file is ignored until the next scheduled run.	Upload
`outDir`	Specify the full path to the directory where the Ariba Data Transfer Tool moves CSV files after it sends copies to Ariba. Default: `set outDir=%topDir%\OutDir`	Upload

Option	Description	Used For
`unsentDir`	When you have an automatic mechanism to create full master CSV files, and it creates a new batch before Ariba Data Transfer Tool has sent the old batch, the old batch does not need to be sent. This option specifies a directory in which to put unsent data. Default: `set unsentDir=%topDir%\UnsentDir` This option is only present in the `masterdata` options file. With incremental data, newer and older files may contain different data, so all files are sent, and this option is absent.	Upload (master data only)
`logDir`	Specify the full path to the directory where the Ariba Data Transfer Tool writes a log file after it finishes the task that uses this options file. Default: `set logDir=%topDir%\UnsentDir`	Upload, Download
`notifyonsuccess`	Specify 1 to enable sending an email message when the data transfer is successful. Specify 0 to disable this feature. Default `set notifyonsuccess=1`	Upload, Download
`notifyonfailure`	Specify 1 to enable sending an email message when the data transfer is not successful. Specify 0 to disable this feature. Default `set notifyonfailure=1`	Upload, Download
`notificationSuccess Subject`	Specify the subject of the notification email message when the data transfer is successful. For example: `set notificationSuccessSubject=AribaFileTransfer Master Data Upload: Success`	Upload, Download
`notificationFailure Subject`	Specify the subject of the notification email message when the data transfer fails. For example: `set notificationFailureSubject=AribaFileTransfer Master Data Upload: Failed`	Upload, Download
`proxyHost`	Host name or IP address of the proxy server. Example: `set proxyHost=localhost`	Upload Download
`proxyPort`	Port used by the proxy server to handle secure communication. Example: `set proxyPort=8081`	Upload Download

Option	Description	Used For

`proxyAuthMod`	Indicates the method used for authentication on the proxy server. If the option is not defined, then no authentication is used with the proxy server.
	
	Note that NTLMv2 is not supported.
	
	Possible values:
	
	`B: basic`
	
	`D: digest`
	
	`N: NTLM`
	
	For example: `set proxyAuthMode=D`
		Upload
		Download

`proxyUser`	The user name used for authentication. Required for all three authentication methods. For example: `set proxyUser=`*`username`*	Upload Download
`proxyPassword`	The password used for authentication. Required for all three authentication methods. This password can be encrypted in the same way you encrypt the shared secret. For more information, see Ariba Integration Toolkit Guide. For example: `set proxyPassword=`*`password`*	Upload Download
`proxyDomain`	The domain to which the user identified by the `proxyUser` option belongs. Required for NTLM authentication only. For example: `set proxyDomain=ARIBA`	Upload Download
`clientKeyStore`	The path where the certificate is stored.	Upload Download
`clientKeyStorePassword`	The password to access the certificate from the key store.	Upload Download
`clientKeyPassword`	The password to access the certificate from the key.	Upload Download

Related Information

Configuring integration toolkit security [page 13]

Troubleshooting the SAP Ariba integration toolkit on Oracle Fusion Middleware

When you encounter an error, you can view the logs associated with the scheduled task and take corrective steps to resolve the error.

SAP Ariba integration toolkit on SAP NetWeaver PI

In this section:

About the SAP Ariba integration toolkit on SAP NetWeaver PI

This feature is available for:

- SAP Ariba Buying and Invoicing
- SAP Ariba Buying
- SAP Ariba Invoice Management
- Ariba Network (Ariba Payment Integration Toolkit)
- SAP Ariba Contracts

The Ariba Network Adapter for SAP NetWeaver allows you to exchange transactional data between Ariba Network and your SAP ERP system. If you are using SAP Ariba Invoice Management for invoice automation, then invoices from Ariba Network are sent to SAP Ariba Invoice Management. To ensure that invoices from Ariba Network are reconciled in SAP Ariba Invoice Management, you need to download master data corresponding to these invoices from your SAP ERP system.

If you use the SAP Ariba integration toolkit to download master data, then you are required to maintain and monitor two different environments, one for master data and the other for transactional data integration. This is cumbersome and time-consuming. Therefore, instead of using the SAP Ariba integration toolkit as a standalone tool, you can now deploy the SAP Ariba integration toolkit on SAP NetWeaver.

This provides the following benefits:

- You have a single solution with SAP NetWeaver PI for both master and transactional data integration.
- You can leverage SAP NetWeaver PI's monitoring, transaction tracking, and error handling capabilities.

You can also use the SAP Ariba integration toolkit on SAP NetWeaver to:

- Enable file-based integration of master and transactional data between SAP Ariba procurement solutions and your SAP ERP system.
- Post payments from your SAP ERP system to Ariba Network.

SAP Ariba integration toolkit can be deployed on:

- SAPNetWeaver PI 7.1
- SAP NetWeaver PI 7.3
- SAPNetWeaver PI 7.4

The server and client certificates that the SAP Ariba integration toolkit (on SAP NetWeaver PI) uses to connect to an SAP Ariba cloud solution must be stored in the SAP NetWeaver PI keystore. Therefore, when a request is sent from SAP Ariba integration toolkit to an SAP Ariba cloud solution, SSL handshake is achieved using the certificates stored in SAP NetWeaver PI keystore.

Enabling the use of the SAP Ariba integration toolkit on SAP NetWeaver PI

Procedure

1. Download the `aribanwfiletransfer.zip` file from connect.ariba.com as follows:
   - Login to connect.ariba.com and on the **On-Demand** tab, click **SAP Ariba Invoice Management** > **Ariba Integration Toolkit for SAP NetWeaver**.
   - SAP Ariba Buying solutions customers - Login to connect.ariba.com and on the **On-Demand** tab, click **SAP Ariba Buying and Invoicing or SAP Ariba Buying** > **Ariba Integration Toolkit for SAP NetWeaver PI**.
2. Save the `aribanwfiletransfer.zip` on your file system.
3. Extract the `aribanwfiletransfer.zip` to an `<InstallRoot>` on the file system. There are two zip files available `aribanwfiletransfer_PI7.1.zip and aribanwfiletransfer_PI7.3.zip` for SAP NetWeaver PI 7.1, and SAP NetWeaver PI 7.3/7.4 respectively. Extract the one that is relevant for you.

   > **Note**
   >
   > Netweaver 7.1 users need to take the zip file which is compiled in JDK1.5 (aribanwfiletransfer_PI7.1.zip).

   The following folders and files are available after the extract:
   - Bin
   - classes
   - META_INF
   - `ariba_JobBean.sca`

   The Bin folder contains the `encryptcustomerkey` and `encryptstring .bat` and `.sh` files. Based on your operating system, you can encrypt your authentication credentials.
4. If your version of SAP netweaver PI is prior to PI 7.3 EHP1, use the Java Support Package Manager (JSPM) tool to deploy the SCA file on SAP NetWeaver PI. From SAP netweaver version PI7.3 EHP1 onwards you must use the SAP Software Update Manager (SUM) tool to deploy the SCA file.

   For information on installing the SCA file or upgrading using the:
   - JSPM, see the *About Deploying the cXML Adapter* section in the *Ariba Network adapter for SAP NetWeaver Setup Guide*.

- SUM tool, contact your SAP systems administrator.

5. Create a View in SAP NetWeaver PI keystore and import the server certificate into that view. You need to import the client certificates only if you need to support client based authentication.
6. Create a scheduled task corresponding to the job you want to run and specify the view name that contains the server and client certificates.

Related Information

Encrypting Your Passwords [page 131]
Managing the SAP Ariba integration toolkit on SAP NetWeaver PI [page 102]

Using the SAP Ariba integration toolkit on SAP NetWeaver PI

The SAP Ariba integration toolkit can be deployed as a set of job definitions in SAP NetWeaver PI and you can configure the execution of the jobs through the SAP NetWeaver PI Jobs Administration. The following job definitions are made available in SAP NetWeaver PI as part of the deployable:

- Full Download
- Full Upload
- Incremental Download
- Incremental Upload
- AribaANUpload
- AribaANDownload
- SAP Ariba Spend Analysis Upload
- SAP Ariba Spend Analysis Download

Now, the SAP Ariba integration toolkit connects to an SAP Ariba cloud solution by using certificates that are stored in the SAP NetWeaver PI Keystore as explained in the graphic below:

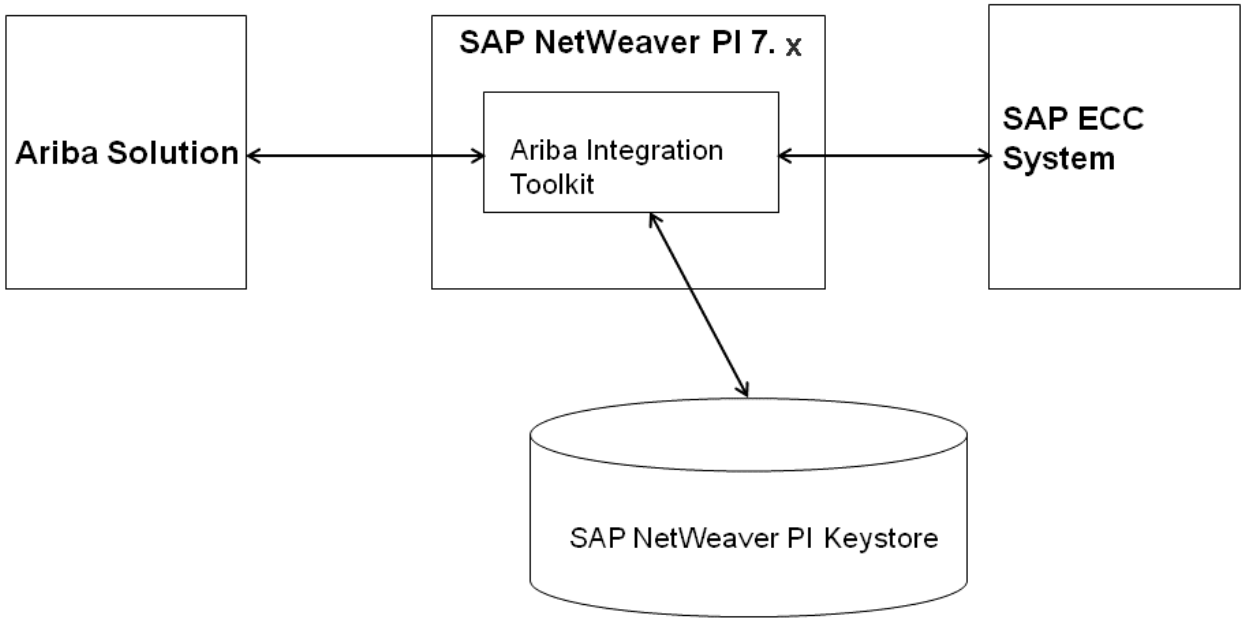


Figure 5: SAP Ariba integration toolkit and the SAP NetWeaver PI Keystore

Managing the SAP Ariba integration toolkit on SAP NetWeaver PI

In this section:

Deploying the SAP Ariba integration toolkit on SAP NetWeaver PI

Procedure

1. Log on to the SAP NetWeaver PI system.

2. Deploy from the `<PI_INSTALL_ROOT>\j2ee\deployment\scripts> directory:`

   Deploy `<user>:<password>@<host>:<port> <path to the installable>/ariba_JobBean.ear`

   > **Note**
   >
   > To undeploy the EAR file, run the command - `undeploy name=ariba_JobBean vendor=ariba.com on_undeploy_error=stop`

Validating the Deployment of the SAP Ariba integration toolkit

Procedure

1. In SAP NetWeaver PI Administrator, click on the **Operations** tab.
2. Click **Jobs - Java Scheduler** tab.

   The following job definitions must be available:

   - AribaIntegrationEventFullDownload
   - AribaIntegrationEventFullUpload
   - AribaIntegrationEventIncrementalDownload
   - AribaIntegrationEventIncrementalUpload
   - AribaANUpload
   - AribaANDownload
   - AribaSpendVisibilityUpload
   - AribaSpendVisibilityDownload

   The following are the optional parameters that are available to all the job definitions mentioned above.

   - useSharedLocation
   - sharedLocationMachineName
   - sharedLocationUser
   - sharedLocationPassword

   > **Note**
   >
   > If SAP NetWeaver PI is running on Windows, then the shared drive option is required because accessing the shared drive in Windows requires NTLM authentication. If SAP NetWeaver PI is running on Linux and you want to use the above mentioned parameters, then the shared location to store the download and upload files is not required because mounted drives which are shared in Linux does not require authentication. On Linux, you can mount a shared drive and configure it with the ease of configuring a local location.

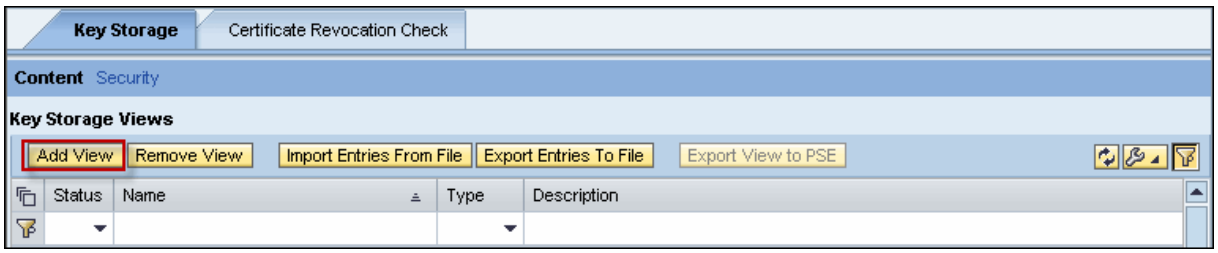
Creating a View and Importing Certificates into SAP NetWeaver Keystore

In this section:

Creating a View in SAP NetWeaver PI Keystore

Procedure

1. Log on to SAP NetWeaver PI Administrator.
2. Click the **Configuration** tab and click **Certificates and Keys**.
3. In the **Key Storage** tab, click **Add View**.

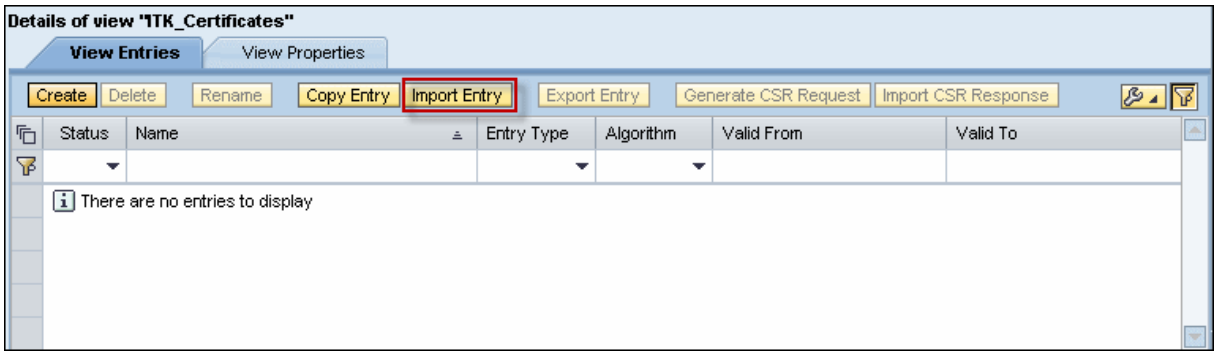


4. Enter a name and description for the view and click **Create**.

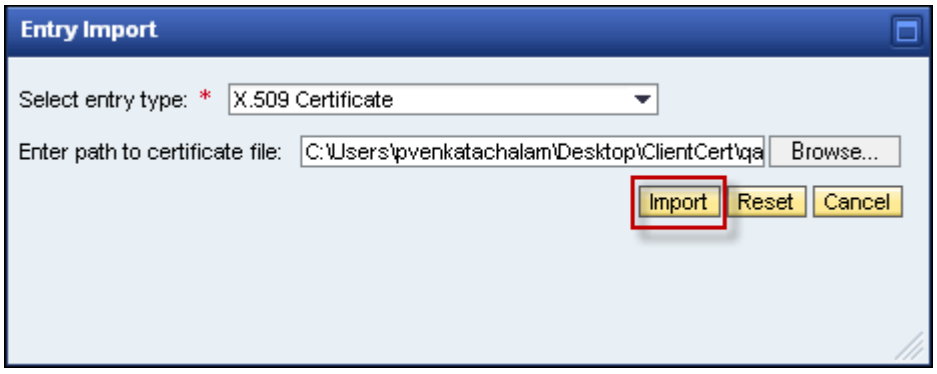
Importing Server and Client Certificates into a View

Procedure

1. In the **View Entries** tab, click **Import Entry**.



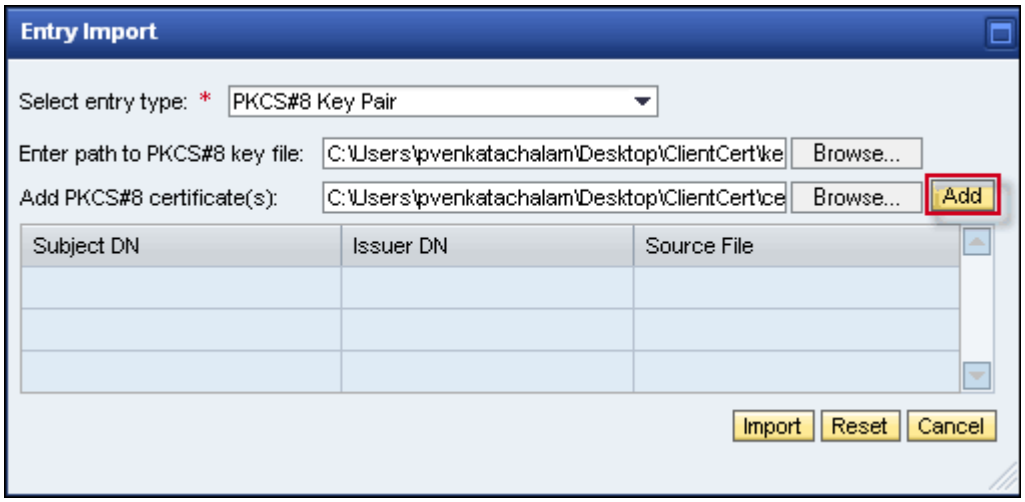
2. Click **Select entry type** pull-down and select **X.509 Certificate**.
3. Enter the path to the location of the server certificate.



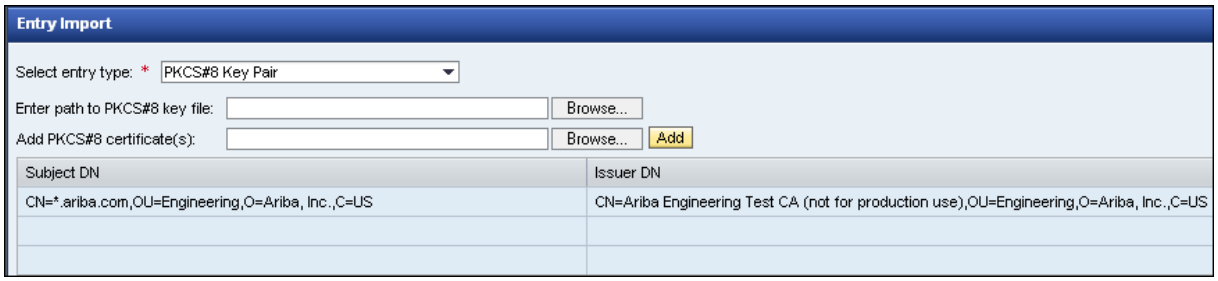
4. Click **Import**.

> **Note**
>
> To import the client certificate, repeat steps 1 to 4 above.

5. To import the key pair, in the **View Entries** tab, click **Import Entry** again.
6. Click **Select entry type** pull-down and select **PKCS#8 Key Pair**.
7. Enter the path to the location of the key file.
8. Enter the path to the location of the client certificate.



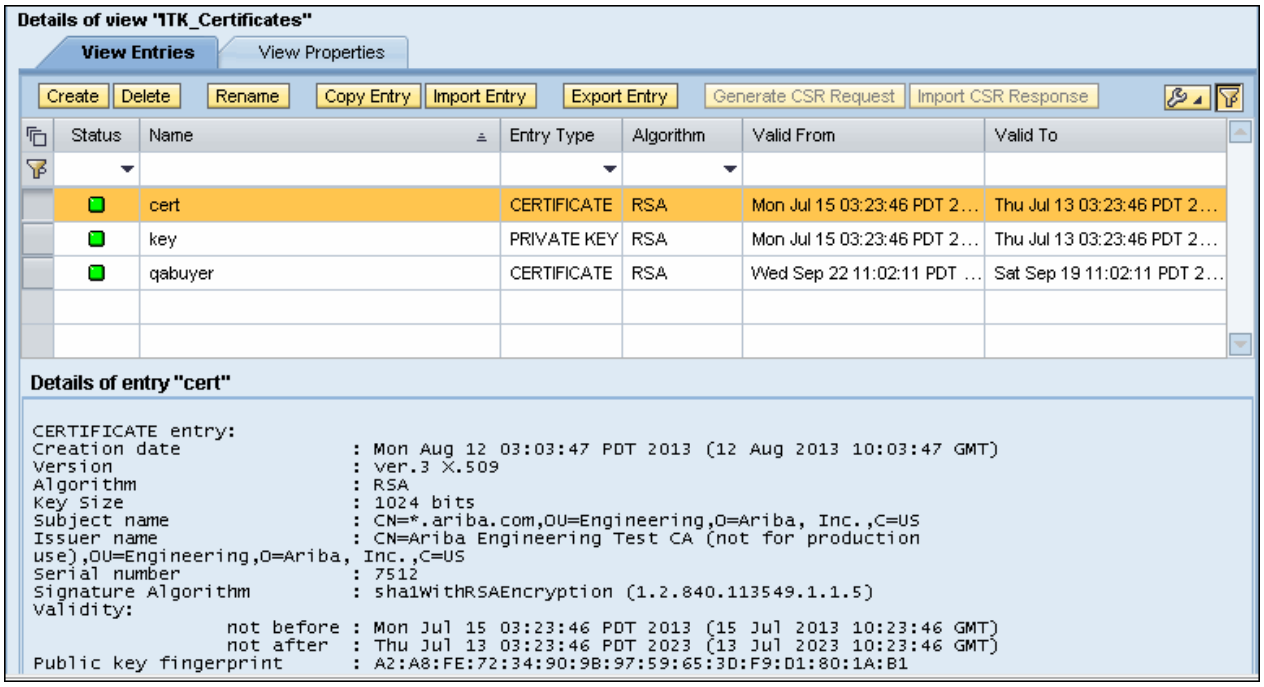
9. Click **Add**.



10. Click **Import**.

Results

The following graphic displays the details of the certificates imported into the `ITK_Certificates` view.

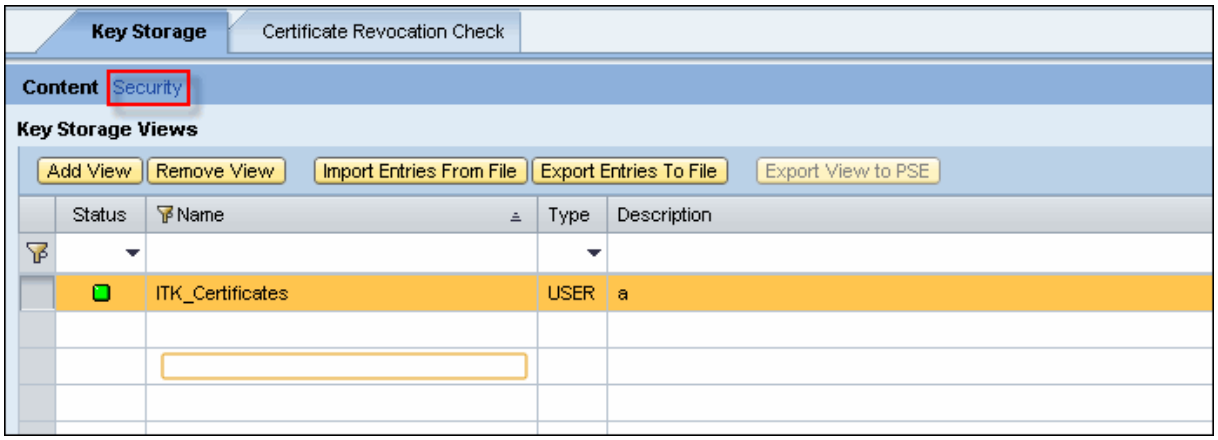


> **Note**
>
> In the above graphic, the server and client certificates are imported into the same view. However, you can have different views for server and client certificates.

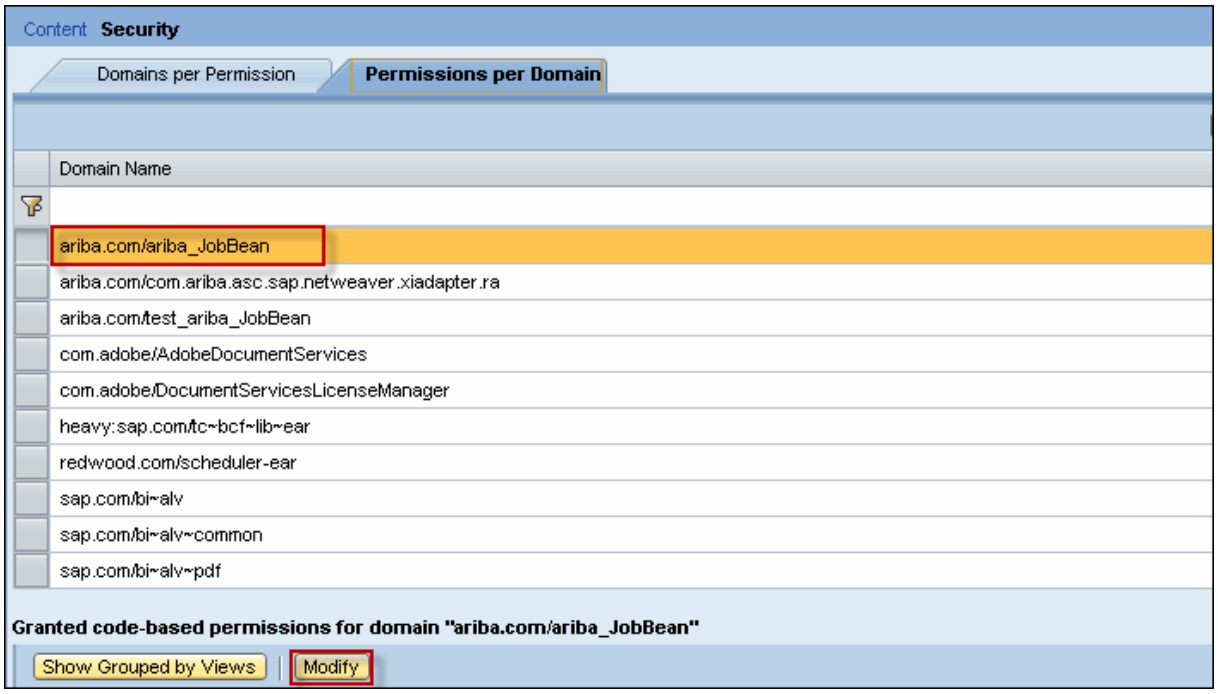
Granting Permissions for the Keystore

Procedure

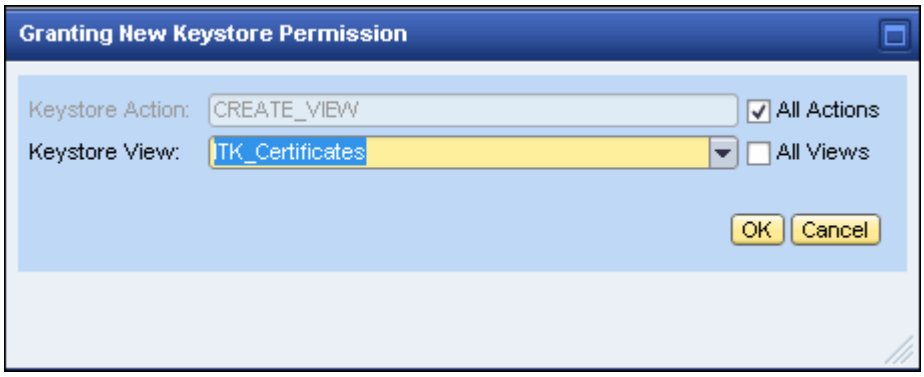
1. In the **Key Storage** tab, select the view for which you want to assign permissions.
2. Click the **Security** link next to the word **Content**.



3. Click the **Permissions per Domain** tab.



4. Select **ariba.com/ariba_JobBean** and click **Modify**.
5. Click **Grant New Permission**.



6. Select **All Actions** check box.
7. In the **Keystore View** field, select the view name and click **OK**.

Creating a Scheduled Task for a Job

Context

You can schedule a Recurring, Cron, or Simple scheduled task. A simple task is executed only once.

Procedure

1. Log on to SAP NetWeaver PI Administrator.
2. Click on the **Operations** tab.
3. Click on **Jobs - Java Scheduler**.
4. Click the **Tasks** tab.
5. Click **Add**.
6. Select the relevant job definition and click **Next**.
7. Enter a name and description for the task.
8. In the **Retention Period** field, enter the number of days.

   Retention period is the number of days the metadata of this task is stored by SAP NetWeaver PI.
9. Enter values for all mandatory parameters. To use certificate based authentication, provide values for new parameters, **truststoreView** and **clientKeystoreView**. For **truststoreView** parameter, enter the name of the view that contains the server certificate. For **clientKeystoreView**, enter the name of the view that contains the client certificate and key file. The view name is case sensitive.

   > **Note**
   >
   > If you are using the secret shared for client authentication, you need not provide a value for the **clientKeystoreView** parameter. However, for the **truststoreView** parameter, you must provide the name of the view that contains the server certificate irrespective of the type of client authentication.
   >
   > To encrypt the shared secret, you must use the `encryptstring.bat` or `encryptstring.sh` from the command line. The encrypted shared secret and key can then be used as parameter values. For more information, see the section Encryption for Security on Page 97 in the Ariba On Demand Solutions Integration Toolkit guide.
10. Click **Next**.
11. Specify the scheduling information for the task and click **Finish**.

Related Information

Scheduled Task Parameters [page 110]

Scheduled Task Parameters

Option	Description	Used For

`urlPrefix`	Examples values are:
	Ariba Procurement solution - `https://s1.ariba.com/Buyer`
	Ariba Sourcing - `https://s1.ariba.com/Sourcing`
	Ariba Network - https://service.ariba.com/ClientToolIntegrator.aw/ad/clienttool
	Using Digital Certificate-Based authentication, prefix the URL with the word `cert` as follows:
	`https://certs1.ariba.com/Buyer`
	`https://certs1.ariba.com/Sourcing`
		Upload, Download, SAP Ariba Spend Analysis

`site`	The name of your company as used in the URL for your Ariba product. The URL is in the format `http://` **`mycompany`**`.procurement.ariba.com.` Example: `set site=mycompany`	Upload, Download, SAP Ariba Spend Analysis
`serviceMode`	Specifies what kind of Ariba product you are using. Possible values include: • `0` to specify an Ariba On-Demand solution. The `urlPrefix` option will, in this case, refer to a site. For example: `http://s1.ariba.com/Buyer/fileupload?realm=realm_1` • `-1` is the default value, which must be changed to the appropriate allowed value to avoid a warning message. The warning message will be issued to the client log, followed by a permanent failure 4xx error code. Site and Service Mode are parameters relevant for jobs related to Ariba procurement solutions and Ariba Sourcing. ◦ AribaIntegrationEventFullDownload ◦ AribaIntegrationEventFullUpload ◦ AribaIntegrationEventIncrementalDownload ◦ AribaIntegrationEventIncrementalDownload For AribaANUpload and AribaANDownload jobs, there are three additional parameters: • Community - The community affiliation of the buyer. • System - The ERP ID configured on Ariba Network for multi-ERP support. • ANID - Specify the ANID of the buying organization. If the buying organization has more than one ERP system, define an application system ID to uniquely identify the ERP system in Ariba SN. For example: set ANID=AN0100000001 set systemID=ERP020	Upload, Download, SAP Ariba Spend Analysis
`encryptionKeyPath`	Specifies the location of the file that contains the encrypted customer key.	Upload, Download, SAP Ariba Spend Analysis

Option	Description	Used For
`sharedsecret`	The shared secret you configured using the Ariba Administrator.	Upload, Download, SAP Ariba Spend Analysis
`event`	Specifies an integration task. To use an integration task with the SAP Ariba data transfer tool, you must configure the task on the server. If you are using Microsoft Windows, you must quote the value of the `event` option. For more information about data import and export tasks, see the import and administration guide for your Ariba solution. For information about the form and contents of CSV files, see Data Dictionary.	Upload (transactional only), Download
`adaptersource`	The value of adapter source can be ALL, EXTERNAL, SYSTEM, or ARIBA MANAGED.	Download
`downloaddir`	Use to specify the directory (named `DownloadFromAriba` by default) containing the downloaded files in directories to be stored for processing by your back-end system. On Microsoft Windows, the directory is created in `C:\TEMP`. On UNIX, it is created in `/var/tmp`.	Download
`timestampFile`	For incremental downloads, specifies the time stamp file that ensures that the download event pulls data saved since the latest successful download.	Download
`filePrefix`	Use to specify a prefix for downloads, for example `expense`, `payment`, or `accruals`. The tool appends a time stamp to downloaded files, indicating when the download was completed.	Download
`notifyemail`	Specify a comma-delimited list of email addresses to which notifications must be sent, if you have enabled notifications for successful or failed data transfers. Example: `set notifyemail=myemail@mycompany.com`	Upload, Download, SAP Ariba Spend Analysis
`smtpMailServer`	Specify the Simple Mail Transfer Protocol (SMTP) server that the Ariba Data Transfer Tool email client uses to send notification messages. Example: `set smtpMailServer=mail.mycompany.com`	Upload, Download, SAP Ariba Spend Analysis
`topDir`	Specify the full path to the directory where the various input and output subdirectories must go. Example: `set topDir=C:\TEMP\FullMasterUpload`	Upload
`inDir`	The folder in which data files are placed for upload to Ariba. Specify the full path to the subdirectory containing the CSV files to be uploaded. The tool expects to find a subdirectory containing the CSV files to be transferred. Example: `set inDir=%topDir%\InDir \your_subdirectory`	Upload

Option	Description	Used For
`waitTimeBetween IncUpload`	During incremental upload, the Ariba Data Transfer Tool (DTT) transfers data from the `InDIR` in a sequence. This ensures that after the DTT transfers data from the first folder to Ariba procurement solution, it starts to move the data from the next folder. This process is repeated till all the data in the `InDIR` is processed. However, when DTT starts to move the data from a folder, if data from the previous folder is still in processing in Ariba procurement solution, a server busy is reported. To ensure that such an error does not occur, you can use the `waitTimeBetweenIncUpload` parameter. The value for this parameter must be set it milliseconds. If you have set this parameter in the batch file, the DTT after sending data from a particular folder, will wait for the configured time interval before it starts to process the data from the next folder.	Upload
`filterprefix`	Specify the subdirectory name prefix that indicates which CSV file subdirectories (in `inDir`) the tool must ignore when uploading files. Default: `set filterprefix=processing` If you have a program that generates CSV files, it must use a file name that begins with this prefix and then rename the file when the file is ready for transfer. If this task is scheduled to run while the file is still being generated, the file is ignored until the next scheduled run.	Upload
`outDir`	Specify the full path to the directory where the Ariba Data Transfer Tool moves CSV files after it sends copies to Ariba. Default: `set outDir=%topDir%\OutDir`	Upload
`unsentDir`	When you have an automatic mechanism to create full master CSV files, and it creates a new batch before Ariba Data Transfer Tool has sent the old batch, the old batch does not need to be sent. This option specifies a directory in which to put unsent data. Default: `set unsentDir=%topDir%\UnsentDir` This option is only present in the `masterdata` options file. With incremental data, newer and older files may contain different data, so all files are sent, and this option is absent.	Upload (master data only)
`logDir`	Specify the full path to the directory where the Ariba Data Transfer Tool writes a log file after it finishes the task that uses this options file. Default: `set logDir=%topDir%\UnsentDir`	Upload, Download, SAP Ariba Spend Analysis
`notifyonsuccess`	Specify 1 to enable sending an email message when the data transfer is successful. Specify 0 to disable this feature. Default `set notifyonsuccess=1`	Upload, Download, SAP Ariba Spend Analysis
`notifyonfailure`	Specify 1 to enable sending an email message when the data transfer is not successful. Specify 0 to disable this feature. Default `set notifyonfailure=1`	Upload, Download, SAP Ariba Spend Analysis

Option	Description	Used For
`notification SuccessSubject`	Specify the subject of the notification email message when the data transfer is successful. For example: `set notificationSuccessSubject=AribaFileTransfer Master Data Upload: Success`	Upload, Download, SAP Ariba Spend Analysis
`notificationFailure Subject`	Specify the subject of the notification email message when the data transfer fails. For example: `set notificationFailureSubject=AribaFileTransfer Master Data Upload: Failed`	Upload, Download, SAP Ariba Spend Analysis
`proxyHost`	Host name or IP address of the proxy server. Example: `set proxyHost=localhost`	Upload, Download, SAP Ariba Spend Analysis
`proxyPort`	Port used by the proxy server to handle secure communication. Example: `set proxyPort=8081`	Upload, Download, SAP Ariba Spend Analysis

`proxyAuthMod`	Indicates the method used for authentication on the proxy server. If the option is not defined, then no authentication is used with the proxy server.
	Note that NTLMv2 is not supported.
	Possible values:
	`B: basic`
	`D: digest`
	`N: NTLM`
	For example: `set proxyAuthMode=D`
		Upload, Download, SAP Ariba Spend Analysis

`proxyUser`	The user name used for authentication. Required for all three authentication methods. For example: `set proxyUser=username`	Upload, Download, SAP Ariba Spend Analysis
`proxyPassword`	The password used for authentication. Required for all three authentication methods. This password can be encrypted in the same way you encrypt the shared secret. For example: `set proxyPassword=password`	Upload, Download, SAP Ariba Spend Analysis
`proxyDomain`	The domain to which the user identified by the `proxyUser` option belongs. Required for NTLM authentication only. For example: `set proxyDomain=ARIBA`	Upload, Download, SAP Ariba Spend Analysis
`clientKeystore View`	The name of the view that contains the client certificate and key file. This is optional.	Upload, Download, SAP Ariba Spend Analysis

Option	Description	Used For
`truststoreView`	The name of the view that contains the server certificate. This parameter is mandatory.	Upload, Download, SAP Ariba Spend Analysis

`ANID`	Specify the ANID of the buying organization. If the buying organization has more than one ERP system, define an application system ID to uniquely identify the ERP system on the Ariba Network.
	For example:
	`set ANID=AN0100000123`
	`set systemID=ERP020`
		AN Upload, AN Download

`systemID`	The ERP ID configured on Ariba Network for multi-ERP support.	AN Upload, AN Download
`community`	The community affiliation of the buyer.	AN Upload, AN Download
`useShared Location`	Whether to use shared location or not. Not required for Linux.	Upload, Download, SAP Ariba Spend Analysis
`sharedLocation MachineName`	Machine name on the shared location machine where the drive is shared.	Upload, Download, SAP Ariba Spend Analysis
`sharedLocation User`	Username of the Operating System	Upload, Download, SAP Ariba Spend Analysis
`sharedLocation Password`	Password of the Operating System	Upload, Download, SAP Ariba Spend Analysis
`authUser`	Use to specify the user name.	SAP Ariba Spend Analysis
`password`	Use to specify your password. Your customer administrator can provide a password. Do not specify `password` for master data uploads or transactional data uploads and downloads. Ariba recommends that you encrypt the value for this option using the `encryptcustomerkey` and `encryptstring` utilities.	SAP Ariba Spend Analysis
`sourceSystem`	Use to specify the source system in a site that the file is for or to specify the source systems to include in the export in a comma-separated list. If you do not use this option, the export includes all source systems. Quotes are required for the value of this option in Microsoft Windows.	SAP Ariba Spend Analysis
`fromEmail`	The from address of the notification email. It is mandatory to specify a valid from address.	Upload, Download, SAP Ariba Spend Analysis
`additionalClientParameters`	Use to specify additional client parameters. Use the following syntax to specify the parameter: `-Key Val <value>` Example: `-performCleanup 1 -cleanupFilesOlderThan 7`	Upload, Download, SAP Ariba Spend Analysis

Option	Description	Used For
`additionalServerParameters`	Use to specify additional server parameters that can be sent to the server. Use the following syntax to specify the parameter: `-Key Val true` or `false` Example: `-checkDimensionReference true -checkDuplicateRow true`	Upload, Download, SAP Ariba Spend Analysis
`fileLocation`	Use to specify a directory of your choice. This directory can only contain either a single ZIP file or CSV files but not both together. You can also specify the path directly to a ZIP file or a CSV file.	Upload, SAP Ariba Spend Analysis
`outDir`	Use to specify where the Ariba Data Transfer Tool must place your CSV file subdirectories after an upload operation. The Ariba Data Transfer Tool creates `success` and `failure` subdirectories within the `outDir` directory and moves the data there based on the result of the upload operation.	Upload, SAP Ariba Spend Analysis
`downloadDir`	Use to specify the location where the star schema export file is stored.	Download, SAP Ariba Spend Analysis
`factTables`	Use to specify the fact tables to include in the export in a space-separated list. If you do not use this option, the export includes all fact tables.	Download, SAP Ariba Spend Analysis
`lastModified`	Use to specify that the export will only include data modified after the specified date in the format DDMMYYYY. If you do not use this option, the export includes all data.	Download, SAP Ariba Spend Analysis
`generateSchemaExport`	Use to specify whether or not to start a new star schema export. By default, `generateSchemaExport` is set to False, and the Ariba Data Transfer Tool downloads existing export files from the specified location. If you set this option to **True**, the Ariba Data Transfer Tool starts a new star schema export operation using the following options you specify and then downloads the results: • `lastModified` • `factTables` • `sourceSystem`	Download, SAP Ariba Spend Analysis

Related Information

Configuring SAP Ariba Spend Analysis data load options [page 34]
Configuring integration toolkit security [page 13]

Setting Time-Out Intervals

Context

You can import large amount of data while running the Supplier Profile Export task. To do so, maintain the following parameter to increase the time-out intervals and avoid time-out errors.

Procedure

1. Log on to SAP NetWeaver PI Administrator.
2. Click on the **Operations** tab.
3. Click on **Jobs - Java Scheduler**.
4. Click the **Tasks** tab.
5. Click **Add**.
6. Select the relevant job definition and click **Next**.
7. Enter a name and description for the task.
8. In the **Retention Period** field, enter the number of days.

   Retention period is the number of days the metadata of this task is stored by SAP NetWeaver PI.
9. Enter "additionalClientParameters".
10. Specify a timeout limit in seconds. Enter "-connectionTimeout <timeout limit in seconds>."
11. Save the changes.

Viewing Logs Related to a Scheduled Task

Context

When you encounter an error, you can view the logs associated with the scheduled task and take corrective steps to resolve the error.

Procedure

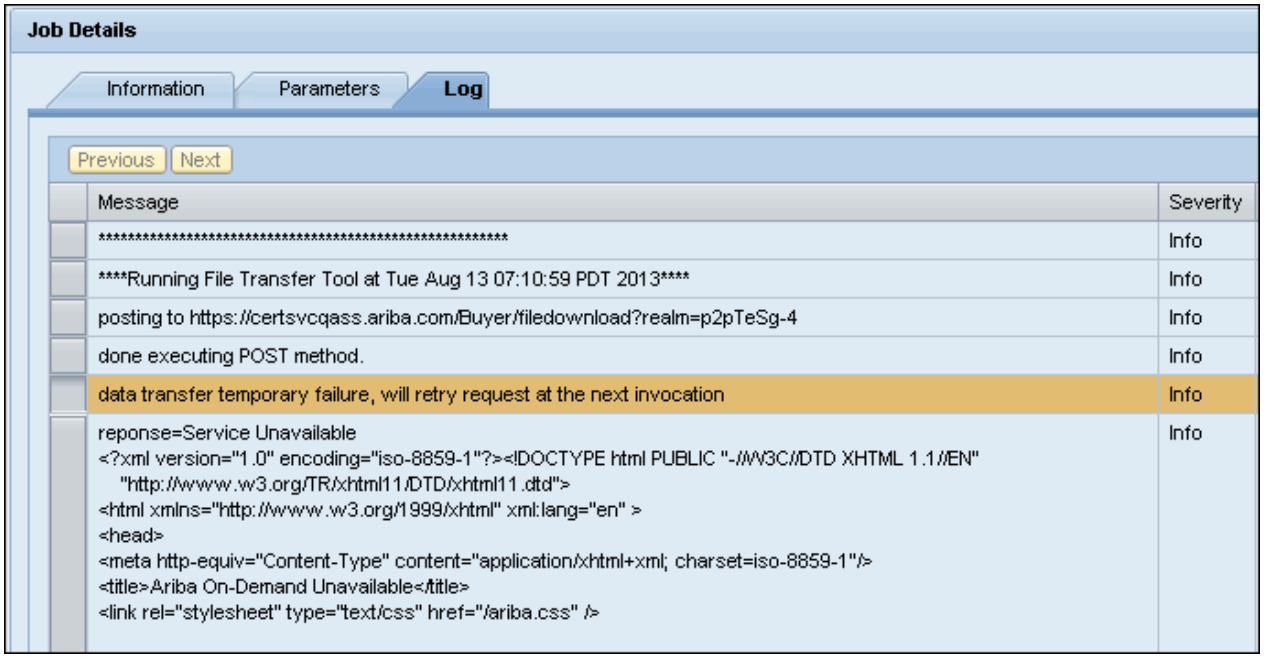
1. Click the **Operations** tab.
2. Click **Jobs.**
3. Click **Java Scheduler**.
4. In the **Jobs** tab, select the relevant job with the ERROR status.

   Under the **Job Details** section, there are three tabs, **Information**, **Parameters**, and **Log**.

5. Click the **Log** tab to view the error details.

Results

The following graphic displays the error details available in `Log` tab.



Related Information

Troubleshooting data transfers [page 66]

Replacing the SAP Ariba data transfer tool with your own tool

In this section:

Audience

This document is for enterprise integration specialists responsible for integrating Ariba On-Demand services with back-end systems.

Overview

The Ariba Integration Toolkit includes the Ariba Data Transfer Tool, which is used to transfer data in batch mode between your back-end system and your Ariba service. Data is transferred in the form of comma-separated-value (CSV) files packaged in a compressed archive.

While the Ariba Data Transfer Tool is a convenient tool to manage data transfers with Ariba, it might not integrate well with your IT environment. You can now replace the Ariba Data Transfer Tool with your own tool, because the underlying protocol used by the Ariba Data Transfer Tool is an HTML form post. A wide range of tools from a simple browser to the most complex Enterprise Application Integration (EAI) tools support this protocol.

The Ariba Data Transfer Tool uses an HTML form-based file upload to transfer data files to the Ariba services. This is the variant of HTML form post designed to transfer files as described in RFC 1867 (http://www.ietf.org/rfc/rfc1867.txt).

This document:

- Describes the information needed to post the HTML form
- Provides reference information for each field in the form
- Gives examples of simple tools you can use to post data
- Provides an inventory of error codes and messages the response can return, along with tools that you can use to troubleshoot the integration

This document applies to the following Ariba solutions:

- The Ariba Procure-to-Pay™ solution package (Ariba Procure-to-Pay)
- The Ariba Procure-to-Order™ solution package (Ariba Procure-to-Order)
- The Ariba Services Procurement™ solution package (Ariba Services Procurement)
- The Ariba Sourcing™ solution package (Ariba Sourcing)
- The Ariba Contract Management™ solution package (Ariba Contract Management)
- The Ariba Supplier Information Management™ solution package (Ariba Supplier Information Management)
- The Ariba Supplier Performance Management™ solution package (Ariba Supplier Performance Management)

Data Flow

The following figure provides a high-level overview of how data flows between your back-end system and the Ariba On-Demand services. It illustrates that any tool capable of uploading a file using a form post can transfer master data. These tools include the Ariba Data Transfer Tool, a browser using an HTML form, and HTTP command line tools such as Curl, and most EAI tools.

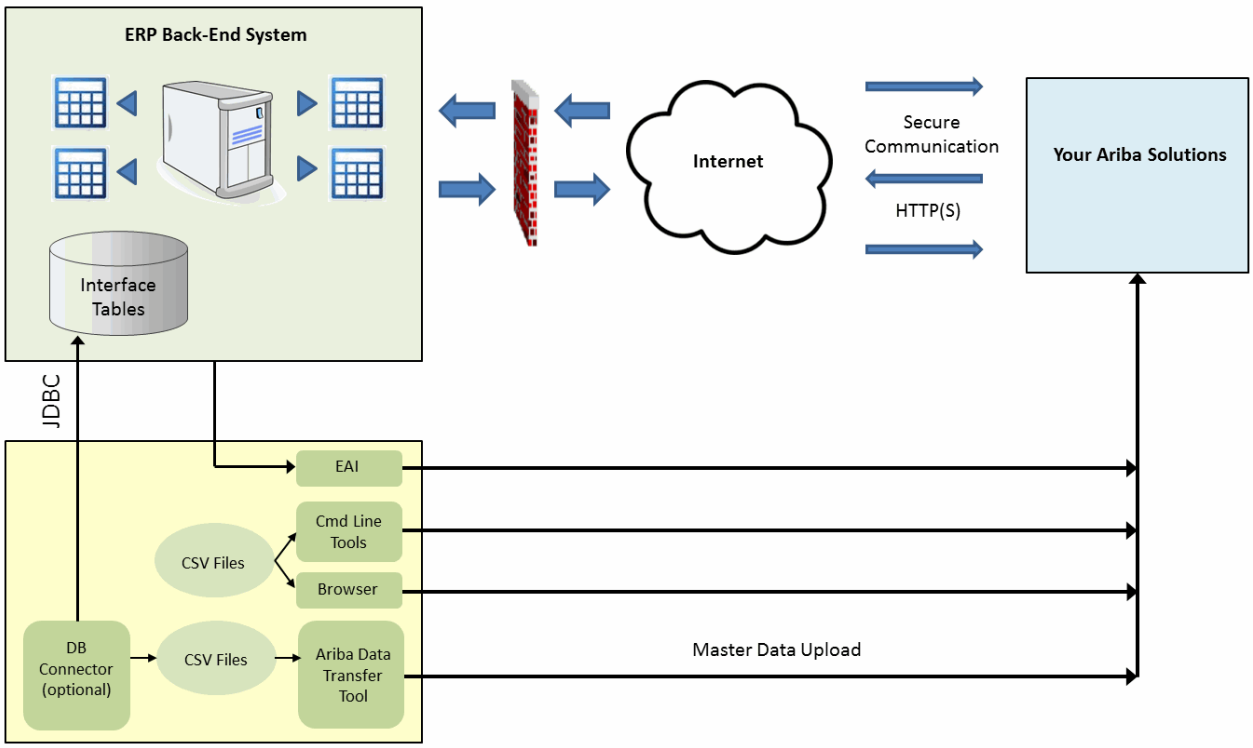


Figure 6: Data Flow Between Back-End System and Ariba On-Demand Services

Prerequisites

- You know the site ID for configuring the URL to which to direct HTTP requests. The Ariba deployment team will provide you with this ID. The URL format is as follows:
  - For Ariba Procure-to-Pay, Ariba Procure-to-Order, and Ariba Invoice and Payment: `https://s1.ariba.com/Buyer/fileupload?realm=YourSiteID`
  - For Ariba Sourcing, Ariba Contract Management, Ariba Supplier Information Management, and Ariba Supplier Performance Management: `https://s1.ariba.com/Sourcing/fileupload?realm=YourSiteID`
- You have configured the integration password. Follow these steps to complete the configuration:
  1. Log in to your site.
  2. Go to Ariba Administrator.
  3. In the left menu, choose **Integration Administration** > **Integration Toolkit Security**.
  4. Enter the integration password twice.
  5. Click **Save**. The integration password is now set, and data transfers using the Ariba Data Transfer Tool or the other tools described in this document are now enabled.
- You are familiar with the files used for loading master data and you have a batch data file available. Also, in the case of procurement solutions, you have uploaded this file by using the Data Import/Export tasks in Core Administration.
- You are familiar with the Ariba integration, import, and administration guides.
- You have read the previous chapters in this Guide.

This document focuses on setting up an alternate tool to the Ariba Data Transfer Tool. It does not provide information on master data itself. For details on what files must be uploaded, please refer to the guides listed above.

Uploading a Master Data File

Uploading the master data file requires you to correctly fill out the parameters in the form post and submit the post to the correct URL. This section provides basic information on the process and gives examples of uploads using available tools such as web browsers and Curl, an HTML command-line tool. It also provides information about implementing a form-based file upload in your EAI tool.

Form Overview

As mentioned earlier, the tool you use to upload data must use an HTML form-based file upload to transfer data files to the Ariba system. The HTML form contains the following information:

- The name of the batch integration event to run
- How to process the master data (that is, incremental load or full load)
- Information on which client is loading the master data (critical for troubleshooting issues when using several clients to upload data onto the server)
- The file to upload

Related Information

Form Parameter Reference [page 124]

Web Browser Example

The easiest way to upload the data is to post the file from a browser. You can try it by loading the following HTML page in your browser. Refer to the instructions that follow the code.

> **Note**
>
> The example below pertains to uploading data to Ariba procurement solutions.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
    <meta http-equiv="content-type" content="text/html;charset=ISO-8859-1">
    <title>Direct access - File upload</title>
</head>
<body>
<h1>Direct access - File upload</h1>
<form method="post"
enctype="multipart/form-data"
action="https://s1.ariba.com/Buyer/fileupload?realm=YourSiteID">
Event: <input type="text" name="event" value="Import Batch Data"><br>
<input type="radio" name="fullload" value="true">Full Load<br>
<input type="radio" name="fullload" value="false" checked>Incremental<br>
Password: <input type="password" name="sharedsecret" value=""><br>
Client Type: <input type="text" name="clienttype" value="Firefox"><br>
Client Info: <input type="text" name="clientinfo" value="Master Data Upload
Form"><br>
Client Version: <input type="text" name="clientversion" value="1.0"><br>
File: <input type="file" name="content"><br>
<input type=submit value="Submit Post">
</form>
</body>
</html>
```

To load data using the preceding sample code:

1. Copy the preceding snippet of HTML code and save it in a file.
2. Open the file with an editor and replace YourSiteID with the actual site (realm) ID.

   > **Note**
   >
   > The URL placeholder in the preceding example applies to Ariba procurement solutions only. The Ariba deployment team provides you with this ID.

3. Open the file with the browser of your choice.
4. Enter the integration password.
5. Provide the path to the batch data ZIP file to upload.
6. Click **Submit Post**.

The master data is uploaded to the server. Upon successful completion of the request, the server returns an HTTP OK (200) with a success message that appears in the browser. The HTTP OK (200) response indicates that a

successful connection has been made, however the master data event is not yet complete. Ariba recommends that you wait for about 10-20 minutes after the HTTP OK (200) response before you schedule the next event to prevent failures.

Related Information

Upload Strategies [page 123]

Curl Example

> **Note**
>
> The example below pertains to uploading data to Ariba procurement solutions.

Curl is a command-line tool for transferring files using a variety of protocols. Curl supports the form-based file upload described in this document. The syntax for uploading a batch data file using Curl is as follows:

```
#!/bin/sh
curl -v  \
          -F "sharedsecret=YourIntegrationPassword" \
      -F "content=@BatchDataFile.zip;type=application/octet-stream" \
      -F "event=Import Batch Data" \
      -F clienttype="Curl" -F clientinfo="OS=Linux:Host=myhost" -F
clientversion="1.0" \
      https://s1.ariba.com/Buyer/fileupload?realm=YourSiteID
```

To execute this command:

1. Copy the preceding snippet of shell script and save it in a file named `runIncrementalUpload.sh`.
2. Open the file with your editor of choice. Replace the integration password, the name of the batch data ZIP file and the site (realm) ID. Save the file.
3. Verify that the batch data ZIP file is in the same directory as the shell script.
4. Run the script: `sh runIncrementalUpload.sh`
   Upon successful completion of the request, the server returns an HTTP OK (200) code with a success message.

Related Information

Upload Strategies [page 123]

EAI Tools

Most EAI tools support form-based file upload. The characteristic of this form is the encoding type set to multipart/form-data.

Each parameter of the form is sent as a part of the multipart stream. Each part must be labeled with:

- A content-type of 'text/plain; charset=US-ASCII' and a content-transfer-encoding of '8bit' for text fields
- A content-type of 'application/octet-stream; charset=ISO-8859-1' and a content-transfer-encoding of 'binary' for file

The MIME parameters section of later in this chapter provides detailed information on the configuration of these parameters in the HTTP request.

Upload Strategies

There are additional considerations that need testing before deploying the adapter in production.

Scheduling

The scheduling of the upload must alternate between incremental and full loads. A typical schedule loads daily incremental updates of master data and refreshes the set of master data weekly. The latter is required because the incremental updates capture only additions and modifications to the existing set, not deletions (deactivations).

The Ariba deployment team can help set up a schedule based on your needs.

Verifying Upload Completion

To verify this data transfer, check with your customer administrator. (Your administrator can verify the upload by checking the status of the appropriate data import/export task in Ariba Administrator. For example, for Ariba procurement solutions, the administrator can check the status of the Import Batch Data task. If the sample data was transferred successfully, the task must have a status of Complete.

Failure and Recovery

The troubleshooting section, explained at the end of this chapter, provides a list of HTTP return codes and corresponding error messages. The return codes provide information on the type of failure and suggest possible reactions from the system. Here is an overview of three types of return codes:

- 200: Successful upload.

- 4xx: Permanent failure.
  The request failed, and either the request or the server must be modified before you resend this request. A typical cause is a mismatch in the integration password. To fix the mismatch, you must either set the password on the server or change the password on the client to match the password on the server.
- 5xx: Temporary failure.
  The server can not process this request at this moment. The client must retry- until the server can process the request. There are a number of reasons for this to happen, for example, several clients are uploading files concurrently, a node is in the process of upgrading itself, or an upgrade of the entire system is in progress.

Related Information

Troubleshooting File Upload [page 127]

Form Parameter Reference

The following table provides reference information for each parameter of the upload, including the parameter name and type, whether it is mandatory, and the possible values it can be set to.

Parameter	Type	Mandatory	Example Value and Notes
`event`	text	Yes	The name of the integration event to run. This event must be a batch event. This is an event that will run other events with the files it finds in the compressed archive. Batch events preconfigured for Ariba Procure-to-Pay are: • Import Batch Data • Import Remittance Data • Import Requisitions • Import Budgets • Export Budget Transactions • Load Budget Adjustments The batch event preconfigured for Ariba Sourcing is: • Import Master Data For information about defining a batch event, see the Chapter2.
`fullload`	text	No	Either true or false. When this parameter is set to false, the data is loaded incrementally. The data must contain the data added or modified since the last upload. When this parameter is set to true, the integration performs a full load. This means that the server will deactivate any data not specified in the uploaded file. Default: false

Parameter	Type	Mandatory	Example Value and Notes
`sharedsecret`	text	Yes	The integration password configured in the Integration Toolkit Security page (on the Integration Manager workspace in Ariba Administrator). The integration password must be configured on the site to enable data upload or download.
`content`	file	Yes	The compressed archive (ZIP file) that contains all the CSV files to upload.
`clienttype`	text	Yes	A string identifying the client uploading the data. This string appears along with runtime information for each integration event run in the Integration Event History page. Example: "WebMethod6.5" To receive email notifications from Ariba for the events run, set clienttype to DataTransferTool in the HTML uploads.
`clientinfo`	text	Yes	A string that contains information about the client environment. This string appears along with runtime information for each integration event run in the Integration Event History page. Example: "JVM 5.0:Windows XP:PurchasinEAI"
`clientversion`	text	Yes	A string that contains the version of the client. This string appears along with runtime information for each integration event run in the Integration Event History page. Example: "1.0"

Master Data Upload Examples

To manage your master data, you frequently load the latest changes, and at a less frequent interval, you fully synchronize the two systems by replacing all master data on the Ariba service with your back-end master data. This section provides examples of parameter values for these operations on Ariba solutions.

Uploads to Ariba Procurement Solutions

The following examples illustrate the two scenarios for uploading master data to Ariba procurement solutions. These incremental and full uploads apply to all Ariba procurement master data batch events.

Incremental Upload

```
curl -v  \
```

```
-F "sharedsecret=YourIntegrationPassword" \
-F "content=@IncrementalMasterData.zip;type=application/octet-stream" \
-F "event=Import Batch Data" \
-F "fullload=false" \
-F clienttype="Curl" -F clientinfo="OS=Linux:Host=myhost" -F clientversion="1.0" \
https://s1.ariba.com/Buyer/fileupload?realm=YourSiteID
```

Full Upload

```
curl -v  \
-F "sharedsecret=YourIntegrationPassword" \
-F "content=@FullMasterData.zip;type=application/octet-stream" \
-F "event=Import Batch Data" \
-F "fullload=true" \
-F clienttype="Curl" -F clientinfo="OS=Linux:Host=myhost" -F clientversion="1.0" \
https://s1.ariba.com/Buyer/fileupload?realm=YourSiteID
```

In the full upload example, the full load flag is set to true, and the ZIP file points to the full extract of the data. Note that in this mode, the import operation executed for each event is determined by the default import option configured in the Ariba system. (Examples of import options are Load, Create, Update Only, and Deactivate.) You can see which import option is the default by opening the corresponding data import task from the Data Import/ Export page in Ariba Administrator.

Uploads to Other Ariba Solutions

The following examples illustrate the two scenarios for uploading master data to Ariba Sourcing, Ariba Contract Management, Ariba Supplier Information Management, and Ariba Supplier Performance Management. These incremental and full uploads apply to all master data batch events for these solutions.

Incremental Upload

```
curl -v  \
-F "sharedsecret=YourIntegrationPassword" \
-F "content=@IncrementalMasterData.zip;type=application/octet-stream" \
-F "event=Import Master Data" \
-F "fullload=false" \
-F clienttype="Curl" -F clientinfo="OS=Linux:Host=myhost" -F clientversion="1.0" \
https://s1.ariba.com/Sourcing/fileupload?realm=YourSiteID
```

Full Upload

```
curl -v  \
-F "sharedsecret=YourIntegrationPassword" \
-F "content=@FullMasterData.zip;type=application/octet-stream" \
```

```
-F "event=Import Master Data" \
-F "fullload=true" \
-F clienttype="Curl" -F clientinfo="OS=Linux:Host=myhost" -F clientversion="1.0" \
https://s1.ariba.com/Sourcing/fileupload?realm=YourSiteID
```

In the full upload example, the full load flag is set to true, and the ZIP file points to the full extract of the data. Note that in this mode, the import operation executed for each event is determined by the default import option configured in the Ariba system. (Examples of import options are Load, Create, Update Only, and Deactivate.) You can see which import option is the default by opening the corresponding data import task from the Data Import/Export page in Ariba Administrator.

Troubleshooting File Upload

The following table lists the possible errors the server can return during a file upload:

HTTP Response Code	Error Message	Comments
400	Invalid Realm	The site ID in the Direct Access URL is invalid. Contact Ariba support to find the correct ID.
400	Must be MIME post with content type <expected string>, received <received string>	The content type of the request is not "multipart/form-data" as expected. Correct the content type to resolve the issue.
400	Illegal form post	• The content disposition of any of the MIME part is not specified or is an empty string, or there is more than one disposition specified. • The name of the MIME part is not specified.
400	Illegal form parameter	The name of the MIME part is not one of the expected parameters. Use the parameters listed in the section "Form parameter reference" earlier in this document.
400	The Ariba Integration Toolkit supports only batch events. Event %s is not a Batch Event.	The event configured on the server side is most likely incorrectly configured or is not a batch event.
400	Cannot find configuration for event: %s	The event passed in the Event parameter is not configured.
400	Illegal type %s	The file is not a compressed archive.
401	Bad credentials -- wrong shared secret	The shared secret configured is null or does not match the one configured on the server.
403	No message	The request is probably configured as a GET instead of a POST.
503	Unable to service realm	An upgrade of the site may be in progress. Retry the request after a few minutes. The header of the response will contain a retry interval attribute with a time in minutes. The default is 5 minutes. The request must be retried after the indicated time.
503	Unavailable to service Event %s: server busy.	An event is already running. This event must complete first before running another one.

HTTPS Request Details

Request Format

The request is an HTTPS Post of a multipart form that contains a set of parameters and a file. Each parameter and the file are sent in separate MIME parts.

The following table describes the format of the HTTPS request:

Parameter	Value
Host	s1.ariba.com
SSL	Required
Method	'POST'
Request URL	"/Buyer/fileupload?realm=YourSiteID" OR "/Sourcing/fileupload?realm=YourSiteID"
Content type	"multipart/form-data"

MIME Parameters

The body of the request contains the parameters listed in the Form Parameter Reference explained earlier. These parameters describe the integration event to run, the credential to use, the file to upload, and information about the client performing the upload.

The following two tables describe how to configure these parameters. Two distinct sets of attributes must be specified for either text or binary parameters.

The following table describes the HTTP attributes required to describe the text parameters:

MIME Part Attribute	Value
content-disposition:	'form-data; name="<parameter name>"'
content-type	'text/plain; charset=US-ASCII'
content-transfer-encoding	'8bit'
Text content	<parameter value>
Content type	"multipart/form-data"

The following table describes the HTTP attributes required to describe the binary parameters:

MIME Part Attribute	Value
content-disposition:	'form-data; name="content"; filename="ct_32732.zip"'
content-type	'application/octet-stream; charset=ISO-8859-1'

MIME Part Attribute	Value
content-transfer-encoding	‘binary’
filename	location of the file on the disk

Sample Request Messages

The following message shows a sample request to upload the file ct_32732.zip to the server. The request includes the following parameters:

```
event:         "Import Batch Data"
sharedSecret:    "yoursharedsecret"
clientInfo:        "10.10.62.98/myvm,Linux,2.6.9-55.0.12.ELsmp,i386,1.5.0_12-
b04,Sun Microsystems Inc."
clientVersion: "1.0"
clientType:     "DataTransferTool"
Content:        "ct_32732.zip"
```

The following message shows the full HTTP post request without the compressed archived:

```
--KsWP7oKcdoVbNxmfgXHe3VZJMRP4pr9_oe4
Content-Disposition: form-data; name="clientVersion"
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 8bit
```

1.0

```
--KsWP7oKcdoVbNxmfgXHe3VZJMRP4pr9_oe4
Content-Disposition: form-data; name="event"
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 8bit
```

Import Batch Data

```
--KsWP7oKcdoVbNxmfgXHe3VZJMRP4pr9_oe4
Content-Disposition: form-data; name="clientType"
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 8bit
```

DataTransferTool

```
--KsWP7oKcdoVbNxmfgXHe3VZJMRP4pr9_oe4
Content-Disposition: form-data; name="sharedSecret"
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 8bit
```

yoursharedsecret

```
--KsWP7oKcdoVbNxmfgXHe3VZJMRP4pr9_oe4
Content-Disposition: form-data; name="clientInfo"
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 8bit
10.10.62.98/myvm,Linux,2.6.9-55.0.12.ELsmp,i386,1.5.0_12-b04,Sun Microsystems Inc.
--KsWP7oKcdoVbNxmfgXHe3VZJMRP4pr9_oe4
Content-Disposition: form-data; name="content"; filename="ct_32732.zip"
```

```
Content-Type: application/octet-stream; charset=ISO-8859-1
Content-Transfer-Encoding: binary
```

Encrypting Your Passwords

Context

You can encrypt your passwords when you deploy Ariba Data Transfer Tool on SAP NetWeaver or Oracle Fusion Middleware.

Procedure

1. Install Java Runtime Environment 1.6, if it is not already installed.
2. Create the options file using the appropriate steps for your operating system.

   For Microsoft Windows:

   a. Using Notepad or a similar text editor, create a new options file called `Transaction.bat`.
   b. In the new file you created, set the `JAVA_HOME` option to the path up to, but not including the bin directory. For example:

      `set JAVA_HOME=C:\Program Files\Java\jre1.x` (the location of your JRE installation)
   c. Save and close the file.

   For UNIX:

   a. Using VI or a similar text editing tool, create a new options file called `transaction.sh`.
   b. Write a line setting the environment variable `JAVA_HOME` to the Java installation directory. For example:

      ```
      export JAVA_HOME=/usr/local/j2re1.6
      ```

      Save and close the file.
3. At the command prompt, change to the directory containing the `encryptcustomerkey` script. For example:

   ```
   cd D:\IntegrationToolkit\bin
   ```
4. Run the script to encrypt the customer key using the following syntax:

   ```
   encryptcustomerkey.bat <options file>
   ```

   where `<options file>` is the path and name of the options file you created where you have set the `JAVA_HOME` option.

   For example:

   ```
   D:\IntegrationToolkit\bin>encryptcustomerkey.bat
   "D:\IntegrationToolkit\transaction.bat"
   ```

   A prompt to enter the key is displayed.
5. Enter the customer encrytion key to be encrypted.

The key must be an alphanumeric string of 24 characters in length. For example, `TestCustomerKey123456789`.

The encrypted key is displayed as an output on the command line. For example,

```
{3DES}+Y9xa4mwCKKvBh6oPRIFvkdk2srHINb/I0JygsfYQWQ=
```

6. Copy the encrypted key from the command line to a .txt file and save the file. For example, `CustomerKey.txt.`

   The encrypted key must be the first and only line in the file without any spaces before or after. The path to this file must be passed as the value for the `encryptionKeyPath` parameter in the options file.

7. In the options file for example, `transaction.bat`, pass the path of the `CustomerKey.txt` file for the `encryptionKeypath` parameter in the options file as follows:

   ```
   <set encryptionKeyPath=D:\IntegrationToolkit\CustomerKey.txt>
   ```

8. Save the options file.
9. Run the script to encrypt a string using the following syntax:

   ```
   encryptstring.bat <options file>
   ```

   where `<options file>` is the path and name of the options file you created where you have set the `JAVA_HOME` option.

   For example:

   ```
   D:\IntegrationToolkit\bin>encryptstring.bat
   "D:\IntegrationToolkit\transaction.bat"
   ```

   A prompt to enter the string is displayed. This string is the shared secret that is configured under Integration Toolkit Security in Ariba Administrator.

10. Enter the string to be encrypted.

    For example: `welcome`

    The encrypted string is displayed as an output on the command line. For example,

    ```
    {3DES}xXk8vlgZl6iVmdCJYy/n1w=
    ```

11. Use this encrypted string to configure your password parameters for SAP NetWeaver and Oracle Fusion scheduled tasks.

Revision history

The following table provides a brief history of the updates to this guide. SAP Ariba updates the technical documentation for its cloud solutions if

- software changes delivered in service packs or hot fixes require a documentation update to correctly reflect the new or changed functionality;
- the existing content is incorrect or user feedback indicated that important content is missing.

SAP Ariba reserves the right to update its technical documentation without prior notification. Most documentation updates will be made available in the same week as the software service packs are released, but critical documentation updates may be released at any time.

Cloud integration release	Month/year of update	Updated chapter/section	Short description of change
9.0	December 2016	N/A	Reset revision history for cloud integration release 9.0.

	January 2017	• Ariba Data Transfer Tool Options
		• Configuring the Ariba Data Transfer Tool
			• Added the following parameters:
			  ○ `numberOfReConnectionsToServer`
			  ○ `waitTimeBeforeReconnection`
			  ○ `waitTimeBetweenIncUpload`
			• Added a note to schedule a minimum interval of 30 minutes between subsequent data transfer events.

	February 2017	Options File Parameters	Added information about support for exporting Contract Line Items Document (CLID) using the integration toolkit.
9.0 SP1	March 2017	N/A	N/A

Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

Accessibility

The information contained in the SAP documentation represents SAP's current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of willful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. All links are categorized for transparency (see: http://help.sap.com/disclaimer).

www.ariba.com