ONESOURCE GLOBAL TAX FOR SAP ARIBA SOLUTIONS

INSTALLATION GUIDE

PRODUCT VERSION 1.0.2.0

Document Version 1

THOMSON REUTERS™

DOCUMENT HISTORY

Version Number	Version Date	Summary
v1	July 25, 2018	First version of this guide

Table of Contents

INTRODUCTION

WELCOME TO ONESOURCE GLOBAL TAX FOR SAP ARIBA SOLUTIONS

Corporations using SAP Ariba Procurement Solutions can improve their tax accuracy by integrating with the ONESOURCE Global Tax for SAP Ariba Solution. This solution enables Buying Organizations to get fast, accurate tax (Sales, Use, GST and VAT like) results on procurement transactions and a complete audit trail from which you can generate both standard and custom reports, as well as information to assist in returns preparations (audit functionality will be part of a future release).

The ONESOURCE Global Tax for SAP Ariba Solutions makes the tax calculation process easy by eliminating the need for preparing the complex tax master data like tax codes, rates, jurisdictions and other look ups manually. The very basic and easy tax master data would suffice in addition to web service configuration to get tax rates from globally recognized, robust tax engine backed up by dynamic tax content.

This solution covers Use Tax, GST and VAT like taxes except taxes for Brazil for SAP Ariba Buying (previous product name: Ariba Procure-to-Order), SAP Ariba Buying and Invoicing. (Previous product name: Ariba Procure-to-Pay), SAP Ariba Invoice Management (previous product name: Ariba Invoice Professional) using SOAP based web services.

WHO SHOULD READ THIS GUIDE?

If you are responsible for installing the solution you should make yourself familiar with this guide. Roles who might need to read this guide are:

- System Administrator
- IT Administrator
- Database Administrator
- ONESOURCE Determination Administrator

Make this guide available to each of these contributors to ensure you have a successful installation.

⚠ Only customer who install the solution on-premise must follow these install instructions. Customers who are managed by Thomson Reuters Cloud or Hosted services don’t need to install the software.

PREREQUISITES

For a seamless and successful deployment of the product the following prerequisites are met:

- Understanding of deployment of Java and Web Application Server based solutions
- Ability to update xml structure files
- Have rights and access to create a new data base schema and connect to it via a JDBC driver
- Installed and configured ONESOURCE Determination
- Access to ONESOURCE Determination with user and role create rights

SYSTEM PREREQUISITES

The following platform is supported for deployment of the Integration for Ariba Solution.

Component	Requirement
Operation System	Red Hat Linux 7.3
Data Base	Oracle DB 12.1
Web Application Server	None - Apache TomCat 8 is embedded in the solution
Java	Java 1.8 JRE
RAM	1 GB min, 2 GB recommended
CPU	No minimum requirement
Disk Space	500MB (logging will take the majority of disk space)

Customers can elect to deploy Integration for Ariba Solutions on the same environment as their other ONESOURCE products are installed (Determination/Reporting, etc.) or setup a dedicated system.

RESOURCES

Resource	Description
Customer Support	Look for answers in the Knowledge Base, or to open a support ticket.
Install Guide	This guide
Configuration Guide	This guide instructs how to configure and setup the integration product to enable tax calculations using ONESOURCE Indirect Tax Determination.

SUPPORT PROTOCOL

The ONESOURCE Global Tax for SAP Ariba Solutions is built, maintained, and owned by Thomson Reuters Tax & Accounting Indirect Tax. The business unit has a dedicated group of employees who have built this product. SAP Ariba built the tax API within SAP Ariba hence they support the process of gathering data within SAP Ariba and making it available to the tax API. Thomson Reuters manages the data mapping from the SAP API to the Determination engine. In case of an issue with the ONESOURCE Global Tax for SAP Ariba Solutions, please follow below mentioned simple steps to open a support ticket with Thomson Reuters:

1. Identify the potential issue and gather all necessary facts (log files, scenarios, configurations, screen prints).
2. Provide steps to reproduce the scenario leading to the issue.
3. Provide system environment information such as your SAP Ariba Procurement Solution version, Cloud Integration version. ONESOURCE Indirect Tax Determination version and ONESOURCE Global Tax Integration for SAP Ariba Solutions integration version.
4. Open a support ticket with Indirect Tax at https://tax.thomsonreuters.com/support/onesource/indirect-tax

INSTALLING INTEGRATION

PRE-INSTALLATION STEPS

Before you can start with the actual installation you will need to perform some preparations. This chapter outlines these steps.

SYSTEM PREPARATIONS

Java Run Time Environment

Your system will need to have Java 1.8 installed, with unlimited security libraries. Information on how to install Java 1.8 can be found here: https://java.com/en/download/help/linux_x64_install.xml. To expand and enhance your Java 1.8 installation with unlimited security libraries, please use the software and instructions found here: http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html.

Oracle database

You will need to have an Oracle database created or available to add a schema to. The sysdba instructions for creating the default "aribaint" schema, for this product, are as follows:

1. Log in as sysdba
2. Create a tablespace for this new schema:
   "create tablespace ariba datafile 'C:\<YOURPATHFORDBFILES>\aribaint\aribaint.dbf' size 500m autoextend on extent management local;"
3. Create a new user schema "aribaint":
   "create user aribaint identified by aribaint default tablespace ariba temporary tablespace temp quota 100m on ariba;"
4. Grant permissions to the new user. Consult with your dba, you may wish to limit permissions.
   "grant dba to aribaint;"

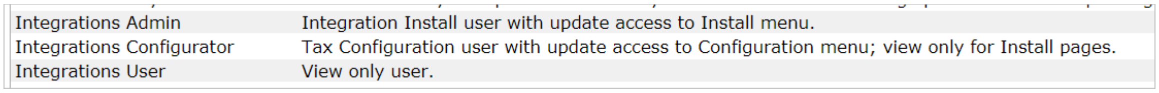
Oracle JDBC Driver

The solution uses the Java Database Connectivity (JDBC) software to communicate with databases. Since this distribution requires the Oracle database, this software requires a copy of the JDBC driver from Oracle that is compatible with this product. Please download a copy of "ojdbc8.jar" from this URL: https://www.oracle.com/technetwork/database/features/jdbc/jdbc-ucp-122-3110062.html.

Determination users and company’s configuration

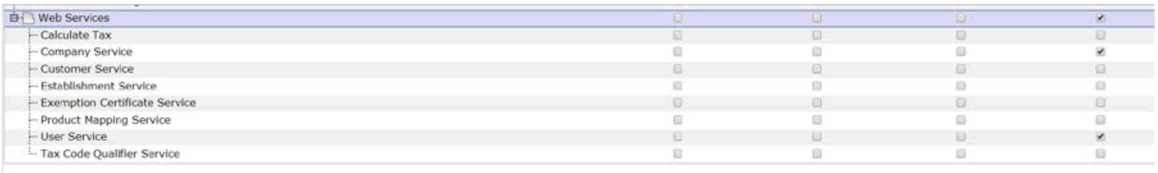
The application depends on and utilizes authentication and authorization services that are built in to the Determination tax engine. You must first set up users, companies, and roles, on Determination, before proceeding with the installation of this integration product.

1. Logon to ONESOURCE Determination as a user administrator or similar
2. Navigate to **Menu > System > Roles**
3. On the top right click on **Actions** button and select **Add**
4. Create the following roles and descriptions:
    a. **Integrations Admin -** Integration Install user with update access to Install menu.
    b. **Integrations Configurator -** Tax Configuration user with update access to Configuration menu; view only for Install pages.
    c. **Integrations User -** View only user.



Integrations Admin	Integration Install user with update access to Install menu.
Integrations Configurator	Tax Configuration user with update access to Configuration menu; view only for Install pages.
Integrations User	View only user.

5. Assign to each role the following permissions by highlighting the role and then selecting the **Edit** tab on the top right. Make sure you assign the following permissions (this will also select the Web Services permission):
    a. Company Service
    b. User Service



Web Services	□	□	□	☑
Calculate Tax	□	□	□	□
Company Service	□	□	□	☑
Customer Service	□	□	□	□
Establishment Service	□	□	□	□
Exemption Certificate Service	□	□	□	□
Product Mapping Service	□	□	□	□
User Service	□	□	□	☑
Tax Code Qualifier Service	□	□	□	□

Role assignment

You will need to assign the newly created users to the appropriate people in your organization. See the Configuration Guide for details on which roles provide what access in the Integration for Ariba Solutions UI.

This ends the system setup and preparations. You are now ready to install the solution.

INSTALLING THE INTEGRATION FOR ARIBA SOLUTIONS

Thomson Reuters provided a zip file for you containing all relevant distributions for the install. Make sure you have the correct version at hand.

This install document assumes a fresh install of the Integration. If you are installing this new version into an environment with our prior version 1.0.1.x Integration already installed please first uninstall the integration. This is a new solution using a new technology.

UNPACKING THE SOFTWARE

The structure of the files in the ZIP distribution package allows easy installation of the product applications. To unpack the ZIP file please follow these steps:

1. Create a top-level deployment folder on your system, e.g., /opt/ONESOURCEAriba
2. Create a user, e.g. "taxadmin", that will own and control the deployed product applications.
3. Log in as that user and copy the ZIP file into the top-level deployment folder.
4. Unzip the deployment ZIP file.
5. Verify that you have the following structure:

```
Docs/
        ProductUpdatesONESOURCEGlobalTaxSAPAribaSolutions_1020.docx
Code/
        ariba-cloud-integration/
                ariba-cloud-integration.jar
                application.yml
                ariba-cloud-integration.conf
        ariba-cloud-integration-v2/
                ariba-cloud-integration-v2.jar
                application.yml
                ariba-cloud-integration-v2.conf
        aria-cloud-ui/
                ariba-cloud-ui.jar
                application.yml
                ariba-cloud-ui.conf
        installer/
                application.yml
                installer.conf
        application.yml
```

CONFIGURING PROPERTIES FILES

YML template files

This product distribution contains files called "application.yml". These files have system properties listed in a key-value syntax known as YAML (see https://en.wikipedia.org/wiki/YAML and http://yaml.org/ for more information). In the distribution files, some of the property values have placeholders inserted, instead of actual values. You will need to replace strings that look like "*{{ DB_PASSWORD }}*", with the actual value that is needed for your installation.

For example, the common YML template file has this property in it:
spring.datasource.username: {{ DB_USER }}

Your system administrator must edit this file, and replace the contents of "*{{ DB_USER }}*" with the actual name of the Oracle schema username, like this:
spring.datasource.username: aribaint

The common YML (spring-boot/application.yml)

There are certain system properties that are used by more than one product application. For example, all four of the applications use the database, so the database connection information is stored in a common YML properties file that all four applications can reference.

1. Ask your DBA to fill in the values for the spring.datasource.url, spring.datasource.username, and spring.datasource.password:

2. Set the determination.service.authentication.iscloud property to have the value "**false**", i.e.:

   determination.service.authentication.iscloud: false

3. If desired, copy the "taxTypeConfiguration" property and values from the ../ariba-cloud-integration/application.yml file, and append to the bottom of this common application.yml file. This property is used by both the ariba-cloud-integration and the ariba-cloud-integration-v2 applications. If you wish to leave the "taxTypeConfiguration" property in the individual integration application.yml files, you do not need to include it in the common application.yml file.

⚠ Above under step 2 you must set the value to **false**. The application won’t be able to autenticate with the on-premise roles you set up in the prior chapter if this value is not set properly.

The installer YML (spring-boot/installer/application.yml)

This file controls how the installer application will be executed:

1. Properties whose values you must leave as is (DO NOT CHANGE):
   liquibase.change-log
   management.context-path

2. Properties whose values you must change:
   logging.level.org.springframework
   logging.level.org.hibernate

   Recommended values for Production installations for these properties is ERROR.
   Recommended values for Test installations for these properties is INFO.

3. Properties whose values you may wish to change:
   server.port - the port number for this JVM application
   actuatorAuth.userName - the login name for looking at spring-boot actuators.
   actuatorAuth.password - the password to log in for spring-boot actuators.

   Additional information on spring-boot actuators can be found here: https://docs.spring.io/spring-boot/docs/current/reference/html/production-ready-endpoints.html

The UI YML (spring-boot/ariba-cloud-ui/application.yml)

This setup controls the UI component of the application:

1. Properties whose values you must leave as is (DO NOT CHANGE):
   determination.userservice.path
   determination.companyservice.path
   management.context-path
   role.configuration.name
   role.install.name
   role.readonly.name
   determination.authorizationservice.path: sabrix/services/userservice/2011-09-01/userservice

2. Properties whose values you must change:
   server.port - replace with a unique port number for this application
   determination.url - replace with a correct URL for the Determination tax engine host.
   all of the "logging.level" properties; these may be removed if not desired.

3. Properties whose values you may wish to change:
   logging.file - the name of the log file
   log.purgeInterval - the time after which XML logs will be deleted from the database (in milliseconds).
   actuatorAuth.userName - the login name for looking at spring-boot actuators.
   actuatorAuth.password - the password to log in for spring-boot actuators.

The Integration YML (v1/v2)

There are two sets of the application.yml files which control the integration web service end-points. They are almost identical, with only two differences:

- name:
- logging.file:

The first file doesn’t have a version indicator, while the second has a v2 indicator. Please make sure you don’t change these settings.

1. Properties whose values you must leave as is (DO NOT CHANGE):

   determination.path

   determination.taxcalc.path

   management.context-path

   taxTypeConfiguration (see 3.1.3 above)

2. Properties whose values you must change:

   server.port - replace with a unique port number for this application (v1 and v2 ports MUST be different)

   determination.url - replace with a correct URL for the Determination tax engine host.

   determination.authorizationservice.path: sabrix/services/userservice/2011-09-01/userservice

   all of the "logging.level" properties; these may be removed if not desired.

   wsdl.server.name - the URL for this server to allow outside view of the WSDL. This can be commented out.

3. Properties whose values you may wish to change:

   logging.file - the name of the log file

   actuatorAuth.userName - the login name for looking at spring-boot actuators.

   actuatorAuth.password - the password to log in for spring-boot actuators.

Configuring Java command-line arguments

In each application folder, a "conf" file is provided that contains the run-time JVM arguments and options available when launching the applications using Java. These settings are optional, and may be modified or removed. Please consult the Oracle Java/JVM documentation for information on each of these settings.

1. Conf file naming convention:

   The ".conf" file takes the name of the application, and uses the extension ".conf". When launching the jar file, the Java Virtual Machine searches for the .conf file and sets its properties according to the values stipulated.

2. Common YML location:

   Each of the .conf files must be edited in order to reference the correct location of the common "application.yml" file. This common file is located in spring-boot/; however, the complete path name must be listed inside each .conf file.

   For example: your installation's top-level folder is called "/opt/ONESOURCEAriba". Your common application.yml file is located at: "/opt/ONESOURCEAriba/spring-boot/". Edit each <appname>.conf file so that it sets the "spring.config.location" property to the correct location.

Creating launch scripts

There are many ways to launch java applications in Linux. Here are some possibilities:

1. Create a unique script for each application. You can create an executable shell script that bundles the following command line instruction into an easy to use script. Here is what might go into an installer.sh script:

   ```
   cd /opt/ONESOURCEAriba/spring-boot/installer; ./installer.jar –
   spring.config.location=/opt/ONESOURCEAriba/spring-boot/ >> /dev/null 2>&1 &
   ```

2. Use a "no hangup" command and launch the jar application in the background. To do this, you can use the following command from the top-level folder:

   ```
   nohup java -jar ./spring-boot/installer/installer.jar &
   ```

3. Referencing the common YML
   Each of the .conf files references a "RUN_ARGS" property, that sets the value of "--spring.config.location". If this is properly edited to point to your installation's path (folder) that contains the common "application.yml" file, your integrations will launch with the common yml properties available to them. See Common YML location in Configuring Java command-line arguments section above.

4. Preparing the Oracle driver
   The Oracle JDBC driver must be available to the Java Virtual Machine on its classpath. Your system administrator can ensure that the ojdbc jar file is placed properly on the Linux server. It is

also possible to simply copy the ojdbc jar file into each application folder, which guarantees presence on the JVM classpath.

LAUNCHING THE APPLICATIONS

Now that all the setup and configuration steps have been performed you can launch the applications. Each application must be started individually and in a certain order:

Running the installer.

When running the installer, there is a special command line argument that will shut down the installer after it completes a successful update of the database. That argument is: "*-DshutdownAfterRunning=true*". Make sure to put this argument in front of the -jar argument, otherwise, the jar executable will launch without capturing this information.

Use a command line or shell script to launch the installer.jar file. Make sure it completes successfully. You may kill the process after it has finished updating the database using liquibase, and the "Application started" log message has occurred.

Launching the UI.

Use the command line, or a shell script, to launch the ariba-cloud-ui.jar file after the installer.jar has executed successfully.

Launching the integration (v1)

Use the command line, or a shell script, to launch the ariba-cloud-integration.jar file after the installer.jar has executed successfully.

Launching the V2 integration.

Use the command line, or a shell script, to launch the ariba-cloud-integration-v2.jar file after the installer.jar has executed successfully.

TESTING AND CONFIRMING YOUR INSTALLATION

Validating the database installation

Log in to the database using sqlplus, SqlDeveloper, or some other tool, with the same schema userName and password that were specified in the common application.yml file.

- Verify that the table DATABASECHANGELOG contains 144 rows. Verify that there is a row whose ID value is "1.0.2.0.115", and that it has a tag value of "1.0.2.0.17".
- Verify that the table DOCUMENT_TYPE has 2 rows.
- Verify that the table US_TAX_SUMMARIZATION_OPTION has 3 rows.
- Verify that the table VAT_SUMMARIZATION_OPTION has 2 rows.

Validating the UI application.

- Open a supported browser, e.g. Chrome.
- Enter the URL for your server's host, along with the "server.port" number specified in the ariba-cloud-UI application.yml file (see Step 3.3.2). This might look like: http://myhostname:8088/
- Verify that the Thomson Reuters ONESOURCE Global Tax for SAP Ariba Solutions login page is displayed.
- Log into the application using a valid username and password as configured in Determination. See additional information about how to use the UI in the Configuration Guide.

Validating the integration application.

- Open a supported browser.
- Navigate to the URL for the ariba-cloud-integration's version actuator, using this format:

  http://<serverhostname>:<integration port #>/actuator/version
- The browser will prompt you for a userName and password. Enter the ones specified in the application's application.yml file.
- Verify that the version number listed is "1.0.2.1.xxx". Where xxx is a three-digit number.
- Repeated above steps for the v2 integration URL.

NEXT STEPS

You have now setup, configured, and started the Integration for Ariba Solutions successfully. Proceed to the Configuration Guide for next steps on how to connect your Ariba Solution.