PeopleFluent

Recruit, Develop, Perform, Reward

PeopleFluent Learning

Rustici Engine Installation Guide

Procedures to install and configure Rustici Engine for SCORM (Single Database)

www.peoplefluent.com/products/learning

Contents

About the Rustici Engine Installation Guide

These instructions are provided for on-premises customers and partners who host the LMS and plan to host Rustici Engine.

About Rustici Engine

Rustici Engine has three required components: a database to store its application data, a file system to store courses, and an application server to serve these courses. Detailed instructions follow for configuring these components.

Audience

This guide is intended for:

- PeopleFluent Learning system administrators who are comfortable with managing application, web and database servers.
- PeopleFluent Professional Services and reseller staff tasked with installing Rustici Engine on-premises.
- PeopleFluent and reseller customer service staff may also find this guide a useful reference to support client system administrators.

This document assumes that you are familiar with the supported operating systems, database servers, web browsers, and network configuration.

Notice to Users

This document is subject to revision based on external hardware and software changes; it will be updated periodically to reflect those changes.

PeopleFluent supports and provides defect fixes for PeopleFluent products under valid Support and Maintenance Agreements on only those operating systems and third-party systems that have been certified and published by PeopleFluent. PeopleFluent will not be responsible for providing any defect fixes for non-certified distributions.

Document Information

This section lists any changes or updates that occur following initial publication.

Table: Document Revision

Revision Information	
Revision Date:	October 26, 2023
Revised Document Version Number:	1.0
Details of Revision:	Initial publication.

System Requirements

Rustici Engine requires PeopleFluent Learning 15.1 or later. To ensure that you have a compatible LMS version, log into the LMS and go to **Manage Center > System > System Statistics > System Activity Statistics**, and verify that the Version property is 15.1.0 or later.

PeopleFluent Learning 15.1 and later require the rustici folder and supporting files, located in the *ekp* folder (for example, \Tomcat*n.n*\webapps\ekp\rustici).

If you are installing the Rustici Engine on your PeopleFluent Learning server, the hardware and software requirements for the LMS are listed in the Example System Configurations section in the PeopleFluent Learning Hardware and Software Requirements.

If you are installing the Rustici Engine on a separate server, see the example configuration below, which is used to support approximately 500 users launching online training concurrently in a managed hosting environment.

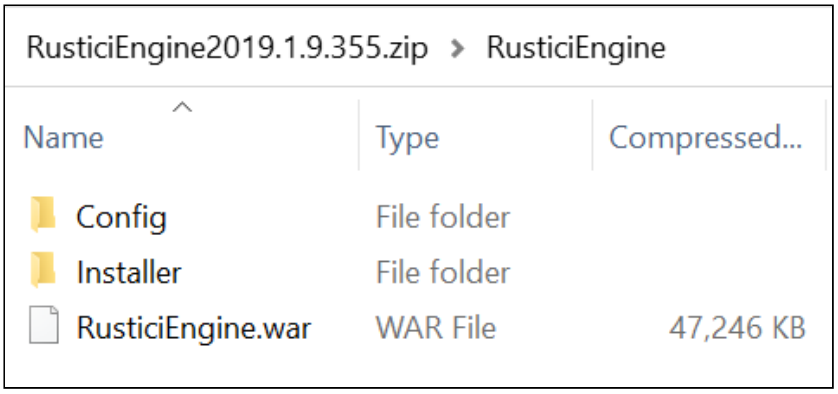
Note that PeopleFluent uses Amazon Web Services EC2 instance types to host PeopleFluent Learning and Rustici Engine. If you need to translate these instance sizes into a hardware spec for on-premises equipment, please refer to the Amazon EC2 Instance Types documentation.

- Concurrent users: 500
- Application Server: t2.medium
    - CPUs: 2
    - RAM: 4 GB
- Database Server: t2.medium
    - CPUs: 2
    - RAM: 4 GB
- Application instances: 2 to 5

Download Rustici Engine

To download Rustici Engine

1. Contact PeopleFluent Support for the RusticiEngine ZIP package.
2. Extract the ZIP file to your server. The ZIP file contains a compressed folder, named RusticiEngine, which contains the WAR file for deployment and the required installer and RusticiEngineSettings.properties configuration file.



RusticiEngine folder

Upgrading to Rustici Engine 2019.1

With the upgrade of Engine to 2019.1, a JSON file named playerConfigurationUrl.json has to be created manually and added to the /<ekp>/player/ folder:

```
{
 "playerConfigurationUrl": "https://<path.to.Rustici.engine>/PlayerConfiguration.jsp",
 "isOffline": false
}
```

Preliminary Considerations

Rustici Engine supports the concept of segmenting data for multiple tenants. As part of this, PeopleFluent supports storing data for all tenants in a single database, or separating out each tenant into its own database (or a hybrid approach with clusters of tenants in separate databases).

By default, Rustici Engine assumes that you are using a single database for all tenants. This is the simplest and most straightforward way to run the engine.

To configure Rustici Engine to use a single database for all tenants, continue reading this section.

To configure Rustici Engine to use a separate database for each tenant in a multi-tenant scenario, see the separate document: *Rustici Engine Multi-Database Configuration*, which is available from the current Release Information page on the PeopleFluent Customer Community.

Database Connectors

For legal reasons, PeopleFluent does not redistribute the database connectors for most platforms, so you may need to download them before continuing.

Rustici Engine requires one of the following JDBC connectors:

- For SQL Server, PeopleFluent recommends the official Microsoft SqlServer JDBC driver.
- For Oracle, download the appropriate driver for your Oracle version and Java version directly from the Oracle website.
- For MySQL, you will need to download MySQL Connector/J.

Once you have downloaded the appropriate driver, copy it to the Installer/lib folder of your Rustici Engine distribution so it will be available for the installer. For example, in Windows, extract the JDBC driver .jar file to **C:\rustici_engine\RusticiEngine\Installer\lib**.

Create the Database for Rustici Engine

Before installing Rustici Engine, you must create its database and an admin user.

For example, to prepare a SQL Server database:

1. In SQL Server Management Studio, create a new database named *rustici*.
2. Create a new login named *rusticiadmin* with SQL Server authentication and non-expiring password.
3. Set the database role membership for the rusticiadmin user to db_owner:
    a. In the Object Explorer, expand **rustici > Security > Users**.
    b. Right-click **rusticiadmin** and select Properties from the context menu.
    c. In the Membership page, select the **db_owner** check box.

Run the Installer Script

Before you run the installation tool, verify that the database connectors required by your installation have been copied to Installer/lib.

Navigate to the *Installer* folder of your Rustici Engine distribution in your command line and run the following command:

```
java -Dlogback.configurationFile=logback.xml -cp "lib/*"
RusticiSoftware.ScormContentPlayer.Logic.Upgrade.ConsoleApp <persistenceEngine>
<connectionString> [schemaPrefix]
```

Where:

- <persistenceEngine> is the name of your DBMS. The value should be either sqlserver, mysql, oracle, or postgresql.
- <connectionString> is the quoted connection string for the database that you want to install Rustici Engine on, followed by a pipe (|) and the class name of your JDBC connector. Ensure that the user specified in your connection string has the appropriate permissions to create and alter any of the necessary tables in Rustici Engine's schema.
- [schemaPrefix] is the name of the schema that Rustici Engine's tables should be created in. This value is optional. If not included, Rustici Engine will use the default schema for the user provided in the connection string.

Example installation commands (without the schemaPrefix specified) for SQL Server, Oracle and MySQL are shown below.

SQL Server

```
java -Dlogback.configurationFile=logback.xml -cp "lib/*"
RusticiSoftware.ScormContentPlayer.Logic.Upgrade.ConsoleApp sqlserver
jdbc:sqlserver://
localhost;DatabaseName=rustici;user=rusticiadmin;password=rust1c1admin^|
com.microsoft.sqlserver.jdbc.SQLServerDriver
```

Oracle

```
java -Dlogback.configurationFile=logback.xml -cp "lib/*"
RusticiSoftware.ScormContentPlayer.Logic.Upgrade.ConsoleApp oracle
"jdbc:oracle:thin:username/password@//192.168.4.231/orcl|oracle.jdbc.OracleDriver"
```

MySQL

```
java -Dlogback.configurationFile=logback.xml -cp "lib/*"
RusticiSoftware.ScormContentPlayer.Logic.Upgrade.ConsoleApp mysql "jdbc:mysql://
myHostName/myDbName?user=myUserName&password=mypassword|com.mysql.jdbc.Driver"
```

The installation command can vary depending on the particular architecture of your Rustici Engine integration. The default instructions above apply to customers who have a single database for their Rustici Engine installation.

Application Server Setup

Rustici Engine for Java needs a Java application container to run and requires Java 8 or later. The following pages describe setting up Rustici Engine for Tomcat. PeopleFluent supports Rustici Engine's use with other Java application containers, but configuring it for use with all Java application containers is outside the scope of this document. Rustici Engine is delivered as a WAR file.

To configure the application server in Windows

1. Copy RusticiEngine.war from the \RusticiEngine folder on your server to \Tomcat9.0\webapps.
2. Allow Tomcat to extract the WAR file to create \Tomcat9.0\webapps\RusticiEngine.
3. Copy the JDBC connector (for example, mssql-jdbc-8.2.0.jre8.jar) to \Tomcat9.0\webapps\RusticiEngine\WEB-INF\lib.
4. Copy RusticiEngineSettings.properties from the \RusticiEngine\Config\ folder on your server to \Tomcat9.0\webapps\RusticiEngine\WEB-INF\classes.
5. Copy rustici.properties from the LMS_2311_Upgrade_Kit_patch_0 folder to \Tomcat9.0\webapps\ekp\WEB-INF\conf.

We specifically recommend that you do not use your Java application container to serve content through Rustici Engine. Instead, you should look at hosting with a traditional web server. There are a variety of ways to map your web server so that the web paths referenced by Rustici Engine are served by a web server rather than the app server serving Rustici Engine.

Required Configuration

Rustici Engine is a highly configurable application. This section describes the settings that are required for almost every installation of Rustici Engine.

While most of these settings have default values that will work for many of our customers, there are a few that must be configured before certain functionality can be used. Rustici Engine can be configured using a couple of different approaches, but the most common and most basic is through your RusticiEngineSettings.properties configuration file.

To add a setting, insert a setting entry anywhere in the root element using the format of `<entry key="SettingName">SettingValue</entry>`.

After adding or changing a setting to your configuration file, restart the Rustici Engine web application to ensure that the change is picked up.

Rustici Engine settings:

- **DataPersistenceEngine**: This configures which DBMS you are using to store Rustici Engine's data. The acceptable values for this setting are sqlserver, mysql, oracle, and postgresql.
- **DatabaseConnectionString**: This is the connection string Rustici Engine uses to communicate with its database. This may be a full connection string or a JNDI resource name.
- **RusticiEngineUrl**: Change the value of this setting to match the host URL for Rustici Engine. You can either specify the entire URL (for example, https://example.com/RusticiEngine), or you can just specify the directory that Rustici Engine is in (/RusticiEngine).
- **ApiBasicAccounts**: To integrate with Rustici Engine's REST API, you will need to configure the accounts that are authorized to use it. The value for this setting must be a newline-delimited list of username/password pairs in the format of username:password. The account credentials are used for Basic Auth to Rustici Engine. You will need at least one account to get started.
- **RedirectOnExitUrl**: This is used to specify where Rustici Engine redirects your learners upon exiting their course. By default, it redirects to a placeholder page that ships with Rustici Engine, but you can replace that value so that it redirects to a specific page within your own application. Like many settings, this can also be configured on a per-tenant, or even per-launch basis if a global value is not desirable.
- **SystemHomepageUrl**: This setting is required if you want to support the use of xAPI courses. It is used in creating certain identifiers in xAPI. The value you specify should be permanent, as changing it can impact data retrieval in xAPI.
- **WebPathToContentRoot**: This setting is required for Rustici Engine to locate course content.

Load Balancer Configuration

For various operations, Rustici Engine will need to construct a fully qualified URL by referencing the incoming request. If you are running Rustici Engine behind a load balancer or a reverse proxy, there are certain issues that can arise if it is not configured properly. Many times in environments like this, the request from the load balancer to Rustici Engine will use an internal domain or an insecure protocol like HTTP. Rustici Engine, not knowing that the request came from a load balancer, will construct a fully qualified URL using this domain and protocol. The resulting URL may be unusable because of the internal domain or the insecure protocol.

To mitigate this scenario, configure your load balancer to pass the original host and protocol of the request via HTTP headers. Rustici Engine looks for and uses different headers when constructing a URL, in the following order of priority:

1. The forwarded HTTP header for RFC7239-style reverse proxying. If present, Rustici Engine checks the host and proto-segments in the value of that header to determine the original host and protocol.
2. Rustici Engine checks for either the *x-forwarded-proto* or *x-forwarded-host* HTTP headers (which were the industry standard way to solve this problem before RFC7239 adoption). If either header is present, it uses the given host and protocol for the original URL.
3. Lastly, Rustici Engine checks for the IIS/ARR-specific header *x-arr-ssl*, and sets the request to be HTTPS if it is found.

Logging

Most logging is handled primarily through SLF4J, an open-source interface to several other logging libraries.

SLF4J can be used in combination with many other logging frameworks. In order to enable a particular framework, you must download the appropriate SLF4J adapter from the SLF4J project website and place it in the classpath for your web application. Rustici Engine already ships with logback, an open-source logging library that includes the SLF4J adapter, so that logging will work out of the box.

A default logback configuration file is also provided, named logback.xml, and is deployed to the WEB-INF/classes directory of your application when the WAR is deployed.

That said, we do not recommend you edit this configuration as your changes may be overwritten when you re-deploy Rustici Engine after a maintenance release. Instead, we recommend you take advantage of the fact that logback.xml is logback's last choice for a configuration file. If you place a logback.groovy or logback-test.xml file in your web application classpath, logback will load that configuration instead. For more information on logback's configuration file format, see the tutorial on the logback project website.

Note that for performance reasons, we do not recommend running with DEBUG level logging enabled in your production environments. The amount of logging involved can degrade performance in some logging configurations.

Synchronize Player Files with Rustici Engine

Starting with PeopleFluent Learning version 22.07, when installing Rustici Engine 2019.1, this section no longer applies. Starting with 22.07, the process described below will no longer be needed as Engine provides an API to download the necessary files.

The integration of PeopleFluent Learning and Rustici Engine includes a set of HTML, JavaScript, and CSS files that enable courses to launch correctly in the learner's browser. Collectively, these are known as the *player files*. These player files must be kept up-to-date with Rustici Engine, as variations even in minor builds of Rustici Engine can impact course behavior.

To ensure the player files remain synchronized with Rustici Engine, a new process was introduced in PeopleFluent Learning 21.04 that enables you to:

- Verify the version of Rustici Engine using the /about API in Engine
- Determine what version of player files are on the LMS server
- Download and update the applicable files from a central location

For partners or customers who host Rustici Engine, there are several steps that need to be included as part of the setup:

1. On the Rustici Engine server locate the *webapps\RusticiEngine\defaultui\player* folder, and zip the folder. Give the ZIP the file name of the full version (for example: *2019.1.zip*).
2. Place the new ZIP file in a location accessible by the LMS server. This can be on the Rustici Engine server, or another server as long as the LMS can access it via either HTTP or HTTPS.
3. In the ekp.properties file for the LMS set the parameter *rustici.playerupdate.url=Domain/Path/* where the player ZIP file is, without the ZIP file name, from step 2 above. The first time this is added, the LMS will need to be restarted.

   PeopleFluent can provide a sample ZIP file, if required.

You must complete steps 1 and 2 each time Rustici Engine is upgraded, but after initially setting the version.txt and ekp.properties files, you can skip step 3.

Troubleshooting

If there are issues launching courses from Rustici Engine, check the ekp.log file. When the error shows *RusticiPlayerUpdateException: About version: '2019.1', Player Version: 'UNKNOWN_VERSION'*, the version.txt file may not exist. Check that version.txt is in the Rustici Engine player folder: **{base}/rustici/player/**.

An example of this error is shown below.

ekp.log Extract

```
2021/May/19 11:06:02.563 ERROR
com.netdimen.txserver.TransactionContainer.handleThrowable: SYSTEM EXCEPTION
(EKP000132093) for user 'jsmith(ekp000007227)', tx 'launch'.
Referring URL: https://lmsserver.com/ekp/servlet/ekp
QueryString:
TX=LAUNCH&SID=EKP000021294&LRNTYPE=O&CID=EKP000004674&EID=EKP000022772&URL=*&openerr
eload=N&ETID=EKP000021294com.netdimen.rusticiengine.player.
RusticiPlayerUpdateException: About version: '2019.1', Player Version:
'UNKNOWN_VERSION'
```

Update ekp.properties to Configure Rustici Engine for SCORM

To take advantage of the enhanced SCORM integration provided by Rustici Engine, you need to set a number of properties in the ekp.properties file.

Version Requirements

- PeopleFluent Learning 15.1 or later
- Rustici Engine 2019.1 or later

Configuration Requirements per LMS Instance

~\WEB-INF\conf\ekp.properties

The ekp.proprties file must include all of the parameters not labelled as optional.

rustici.uri

The fully qualified URL that both the LMS and LMS users can use to access the application root of the Rustici Engine instance via HTTPS.

This is the hostname in the database connection string in the RusticiEngineSettings.properties file. For example:

```
rustici.uri=http://localhost:8080/RusticiEngine
```

You do not need to include the port number (:8080) for Apache https configuration.

rustici.username

Rustici Engine API ID/Username. Use the value from the ApiBasicAccounts entry in the RusticiEngineSettings.properties file.

rustici.password

Rustici Engine API Secret/Password. Use the value from the ApiBasicAccounts entry in the RusticiEngineSettings.properties file.

rustici.enginetenantname

A key/string used to (ideally globally) uniquely identify the PeopleFluent Learning instance to which all respective Rustici Engine communication should be scoped.

Set this to **default** to use the value set in RusticiEngineSettings.properties:

```
rustici.enginetenantname=default
```

rustici.importdefault

This property's value indicates whether the **Use Rustici Engine** checkbox on the Import content package page is checked (true) or unchecked (false), by default.

rustici.manifestjwtduration (Optional)

At course import time, the LMS provides Rustici Engine with a JSON Web Token (JWT)-secured URL to an LMS API that the engine can use to consume the respective course's manifest document(s). This numeric value sets the expiration (in milliseconds) of the JWT. The default Value is 600000 (10 minutes).

rustici.postbackjwtduration (Optional)

At course launch time, the LMS provides Rustici Engine with a JWT-secured URL to an LMS API that the engine can use to pass runtime data about the learner's activity. This numeric value sets the expiration (in milliseconds) of the JWT. The default Value is 18000000 (5 hours).

Caveats

Omitting these properties will cause the LMS to degrade gracefully, and function without Rustici Engine. However, omitting or removing these properties on an instance that has been running with these features enabled is not a supported use-case. Any courses installed with *Use Rustici Engine* checked will not launch or preview correctly, and there may be other undesirable side-effects. If this happens, restore the correct Rustici Engine instance configuration properties in ekp.properties.

If ekp.properties contains invalid configuration properties (or if the LMS is unable to successfully validate these properties via the API), the LMS will not start and users will be unable to access it in a browser.

Network Topology

Rustici Engine must be accessible to the relying LMS instance and its clients, via HTTPS, by the URI defined in the rustici.uri parameter.

Cross-Origin Resource Sharing (CORS)

App CORS policy must be configured to allow requests from users originating from any URL(s) they may use to access any relying LMS instances.

Credentials

Each relying LMS instance must have its own Rustici Engine Tenant created. A set of valid Rustici Engine Tenant-scoped API credentials must be created and configured in the respective relying LMS instance via *rustici.username*, *rustici.password* and *rustici.enginetenantname*.

Update ekp.properties for Watershed Analytics integration

SaaS Customers Only

The Watershed Analytics integration is for SaaS customers only at this time. This information is included in the documentation for informational purposes, so customers hosted by PeopleFluent are aware of settings added to their LMS. On-premise and partner-hosted customers with a Rustici Engine integration are eligible to purchase a full license from Watershed LRS.

Watershed Analytics can deliver near real-time learning activity analytics via dashboards to the LMS, using xAPI statements communicated through Rustici Engine. To enable integration with Watershed Analytics, you must add the following settings to ekp.properties:

```
rustici.watershed.orgid=<{orgId}>

rustici.watershed.url=<watershed url>

rustici.watershed.secret=<{watershed lms secret}>

rustici.watershed.key=lms-<client name>
```

The absence of these properties will disable Watershed—LMS integration. Incorrect or invalid values may prevent startup and produce error logs upon startup.

Update ApiBasicAccounts

Rustici Engine instance properties are configured in the ~\WEB-INF\classes\RusticiEngineSettings.properties file relative to the root of the web application. You can update the properties listed below from their default values to change the behaviour of Rustici Engine.

ConvertTinCanFromScormRealtime

The value of this property should be set to true; This is to ensure Rustici Engine generates xAPI statements for non-xAPI content types.

Example:

```
<properties>
…
<entry key="ConvertTinCanFromScormRealtime">true</entry>
…
</properties>
```

xAPIStatementPipeEnabled

This property is not present by default, but should be added and set to true. While non-essential for the normal operations and/or interactions with PeopleFluent Learning, enabling this feature is necessary in order for outbound statement pipes to function, which will be required for future integration with Watershed.

Example:

```
<properties>
…
<entry key="xAPIStatementPipeEnabled">true</entry>
…
</properties>
```

DataPersistanceEngine

For SQL Server:

```
<properties>
…
<entry key="DataPersistenceEngine">sqlserver</entry>
…
</properties>
```

For the SQL Server example database setup used in this document. Note there is no caret (^) before the pipe (|) when you specifty the connection string in the RusticiEngineSettings.properties file.

```
<properties>
…
<entry key="DatabaseConnectionString">jdbc:sqlserver://
localhost;DatabaseName=rustici;user=rusticiadmin;password=rust1c1admin|
com.microsoft.sqlserver.jdbc.SQLServerDriver</entry>
…
</properties>
```

ApiBasicAccounts

These properties select a username and password for the API accounts. PeopleFluent recommends changing the defaults (test and test).

```
<properties>
…
<entry key="ApiBasicAccounts">test:test</entry>
<entry key="RusticiEngineUrl">/RusticiEngine</entry>
<entry key="RedirectOnExitUrl">/RusticiEngine/defaultui/defaultredirect.jsp?
registrationId=${RegistrationId}</entry>
…
</properties>
```

SystemHomepageUrl

For Apache (https) configuration, exclude the port number after the hostname.

Example:

```
<properties>
…
<entry key="SystemHomepageUrl">https://localhost/RusticiEngine/defaultui/admin/</
entry>
…
</properties>
```

Update WebPathToContentRoot

In the ~\WEB-INF\classes\RusticiEngineSettings.properties file, set the WebPathToContentRoot property to the course content directory.

Example:

```
<properties>
…
<entry key="WebPathToContentRoot">/ekp/nd/fresco/content</entry>
…
</properties>
```

Rustici Engine Post-installation Checks

To validate the ekp.properties and RusticiEngineSettings.properties files

1. Restart Tomcat and check the Rustici Engine webapp starts.
2. Go to **http://localhost:8080/RusticiEngine/defaultui/admin/** and log in with the credentials in the RusticiEngineSettings.properties file.
3. In the LMS, go to **Manage Center > System > Status and Logs > System Log** and verify that there were not errors when logging in.
4. In the LMS, go to **Manage Center > Learning > Import > Import content package** and verify that the Use Rustici Engine check box is either selected or cleared according to the rustici.importdefault setting in ekp.properties.
5. Import a package, enroll in it, launch it, navigate and close it.

Legal Notice

This document has been created for authorized licensees and subscribers ("Customers") of the software products and associated services of Learning Technologies Group, Inc. by its division PeopleFluent and all of its affiliates (individually and collectively, as applicable, "PeopleFluent"). It contains the confidential and proprietary information of PeopleFluent and may be used solely in accordance with the agreement governing the use of the applicable software products and services. This document or any part thereof may not be reproduced, translated or retransmitted in any form without the written permission of PeopleFluent. The information in this document is subject to change without notice.

PEOPLEFLUENT DISCLAIMS ALL LIABILITY FOR THE USE OF THE INFORMATION CONTAINED IN THIS DOCUMENT AND MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO ITS ACCURACY OR COMPLETENESS. PEOPLEFLUENT DISCLAIMS ALL IMPLIED WARRANTIES INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. PEOPLEFLUENT DOES NOT GUARANTEE THAT ITS PRODUCTS OR SERVICES OR ANY SAMPLE CONTENT CONTAINED IN ITS PRODUCTS AND SERVICES WILL CAUSE OR ENABLE CUSTOMER TO COMPLY WITH LAWS APPLICABLE TO CUSTOMER. USERS ARE RESPONSIBLE FOR COMPLIANCE WITH ALL LAWS, RULES, REGULATIONS, ORDINANCES AND CODES IN CONNECTION WITH THE USE OF THE APPLICABLE SOFTWARE PRODUCTS,INCLUDING, WITHOUT LIMITATION, LABOR AND EMPLOYMENT LAWS IN RELEVANT JURISDICTIONS. THE PEOPLEFLUENT PRODUCTS AND SAMPLE CONTENT SHOULD NOT BE CONSTRUED AS LEGAL ADVICE.

Without limiting the generality of the foregoing, PeopleFluent may from time to time link to third-party websites in its products and/or services. Such third-party links are for demonstration purposes only, and PeopleFluent makes no representations or warranties as to the functioning of such links or the accuracy or appropriateness of the content located on such third-party sites. You are responsible for reviewing all content, including links to third-party web sites and any content that you elect to use, for accuracy and appropriateness, and compliance with applicable law.

Any trademarks included in this documentation may comprise registered trademarks of PeopleFluent in the United States and in other countries.

Microsoft, Windows, and Internet Explorer are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Oracle and PeopleSoft are registered trademarks of Oracle International Corporation. Adobe and Acrobat are registered trademarks of Adobe Systems Incorporated. All other names are used for identification purposes only and are trademarks or registered trademarks of their respective owners. Portions of PeopleFluent Workforce Communication software may include technology licensed from Autonomy and are the copyright of Autonomy, Inc.Quartz Scheduler is licensed under the Apache License.

Website: peoplefluent.com