peoplefluent

PeopleFluent Learning

Installation Guide
Release 24.04

Contents

About the Installation Guide

This document provides detailed information about configuring a hosting environment and installing PeopleFluent Learning 24.04.

PeopleFluent Learning has an open standards-based architecture that allows organizations to create, distribute, document and manage training over the internet. It is a Java-based middleware server that works on a variety of platforms and in cooperation with multiple third-party web servers or transaction servers.

Some configuration information is included for Apache, Apache Tomcat, and Microsoft SQL Server. For more detailed installation instructions of third-party services or applications, please refer to vendor-specific documentation.

For more information about the LMS system requirements, see PeopleFluent Learning Hardware and Software Requirements.

If you have any issues or queries related to installing the LMS, please contact the relevant party or parties described in your Technical Support Contract.

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 PeopleFluent Learning 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

Revision Information	
Release	24.04
Revision Date:	April 25, 2024
Revised Document Version Number:	1.0
Details of Revision:	Initial publication.

Prepare the Hosting Environment

PeopleFluent Learning is deployed as a Web Application Resource (WAR) file. If your organization already has established processes and procedures for deploying WAR files in Tomcat, we suggest that you still review this section in detail, as it contains specific information to help you integrate the LMS deployment into your existing processes.

Paths

PeopleFluent recommends locating the Tomcat installation, along with any supporting binaries and configuration files, in a separate partition from your system partition. By convention, we use the following base path in this document for Microsoft Windows environments: **D:\nd**.

Domain

You will need to choose a fully qualified domain name for your installation, in the format *myPeopleFluentLMS.mydomain.com*. This domain name will need to resolve via DNS to an address that points at the application server host for the LMS. This will also be the domain name from which all of your SCORM, xAPI, AICC, and other standards-based content is served. Changing domain names after installation can be challenging and cause considerable disruption.

SSL

To provide secure communications for the LMS deployment, PeopleFluent recommends configuring the use of an SSL certificate. Creation and installation of the certificate is beyond the scope of this document, other than the steps outlined in setting up Apache web server below.

Download PeopleFluent Learning

To download the PeopleFluent Learning installer:

1. Go to http://onlinehelp.netdimensions.com/ and select the LMS version you want to install to go to the LMS Downloads page.
2. If you already have PeopleFluent Learning installed and you are upgrading to a later version, download the Upgrade Kit and refer to the version-specific PeopleFluent Learning Upgrade Guide, which is available from PeopleFluent.
3. If you are not upgrading an existing installation, scroll down to Latest Build Patch and download the patch WAR file.

Secure your License File

The PeopleFluent Account Management team will provide you with a license file for your deployment of PeopleFluent Learning, which will be named *license.ekp*. Ensure that you have this license file to hand, as it will be required later in the installation process.

Install Java Development Kit (JDK) 8

Download either Oracle JDK 8 (with a valid license for use) or OpenJDK (for example, AdoptOpenJDK or Zulu).

Install the JDK to the LMS base path (see **Paths**, above), in a sub-folder that matches the version number (for example, D:\nd\jdk_1.8.0_213). This will facilitate easy and reversible JDK upgrades in the future.

Install Apache 2.4

Base installation instructions and download links are available from: https://httpd.apache.org/docs/current/platform/windows.html

During the setup:

- Place the installation within the base path directory (D:\nd\apache2.4).

- Use the fully-qualified domain name that you selected for your server as the ServerName.

- Select for All Users, on Port 80, and as a Service.

Verify that Apache is working by opening http://localhost/ in web browser. You should see the generic Apache landing page.

Install Tomcat

PeopleFluent Learning 24.04 supports Tomcat version 9.0.64 or later.

> ⓘ PeopleFluent Learning 24.04 does not work with Tomcat 7. Customers and partners who have been using Tomcat 7 must upgrade. See http://tomcat.apache.org/ for instructions.
>
> While previous versions of the LMS have been tested with Tomcat 8.0, this version has reached its end-of-life and is no longer supported by Apache Software Foundation. PeopleFluent advises customers and partners who host their instances of the LMS using Tomcat 8.0 to consider upgrading to Tomcat 9.
>
> Version 24.04 was tested exclusively using Tomcat 9.0.64. You can download Tomcat 9.0.x from https://tomcat.apache.org/download-90.cgi.

During the setup:

- Install Tomcat in the base path directory (D:\nd\tomcat8).

- Select the 'Native' library during the installation.

- Select the path to your installed JDK (see **Install Java Development Kit (JDK) 8**, above.).

- For the Tomcat Administrator Login, set the user name to **ndadmin** and enter a strong password.

Start Tomcat and verify it is working by opening http://localhost:8080/ in a browser. If the generic Tomcat landing page does not open, see Appendix A – Troubleshooting.

After Tomcat is installed:

1. Run the Tomcat Config tool. For Tomcat, for example, this will be in: **D:\nd\Tomcat 9.0.64\bin\Tomcat9w.exe** and set the startup type to Automatic.
2. In the Java tab in the Config tool, set the Maximum Memory Pool to approximately 75% of the available system RAM.

> ⓘ   The Maximum Memory Pool value has a powerful effect on system performance. We suggest a minimum of 6 GB for a production system. A good rule of thumb is that the Maximum Memory Pool size should be set to 75% of available system RAM on a dedicated application server. The remaining 25% of available RAM serves the OS and the Apache web server processes.

Install the LMS

Extract the Application Files

If your organization has an established process for deploying WAR files in Tomcat, you can use it to deploy PeopleFluent Learning. If you do not have an established deployment process, follow the steps below:

1. Create a folder called *ekp* in your Tomcat installation's webapps directory. Following this document's convention, the resulting path would be **D:\nd\Tomcat8\webapps\ekp**.
2. Unzip the contents of the WAR file that you downloaded earlier into this folder. You can use any zip application to unzip the file, for example WinZip or 7-Zip.

Configure Tomcat's Context.xml for Externalized Content

When a WAR file is deployed atop an existing deployment, files in the web application directory are overwritten or (on occasion) removed. To avoid this, you must set up a special directory for your customizations outside of the rest of the system. This process is called content externalization.

To enable content externalization, you add a 'base' parameter to your Tomcat deployment, which PeopleFluent Learning uses to determine the location of custom content:

1. Create a folder in your Tomcat installation's webapps\ekp folder called *content*.
2. Edit the Context section of your Tomcat installation's context.xml file, and include the line below that specifies the parameter named base:

```
context.xml

<Context>

...

   <Parameter name="base" value="D:\nd\Tomcat8\webapps\ekp\content"/>

...

</Context>
```

If the base parameter is unspecified, it is taken to be the same as the web application root directory. You can run PeopleFluent Learning this way, but note that subsequent upgrades may overwrite your customizations. When in doubt, always enable the content externalization base.

PeopleFluent Learning Directory Structure

When the LMS has installed, you can review the application's directory structure to familiarize yourself with file locations.

In the table below, the ${base} parameter refers to a directory that can be external to the web application and should be defined in context.xml as per the above instructions.

Directory	Description
\ekp	The default context of PeopleFluent Learning installed in the web application deployment directory.
\samples	This stores sample files such as skins, an RSS configuration file and certificate PDFs.
\nd\fresco\courses	The default location to install courseware.
\nd\fresco\help	The location of the online help files.
\nd\fresco\txstat	The output directory for the transaction statistics CSV report.
${base}\nd\fresco\content	Stores the extracted self-contained course packages.
${base}\nd\fresco\dataloader	Stores the various data loader files (excluding data loader templates) produced by the LMS.
${base}\nd\fresco\repository	The directory for the repository manager.
${base}\nd\fresco\batchreports	Location for storing batch reports produced by the LMS.
${base}\nd\fresco\styles	The location of installed skins. On fresh installation, the folder would be copied to the configured location as it contains default skins.
${base}\WEB-INF\indexes	Apache Lucene search index files.

${base}\WEB-INF\conf	The directory storing customized login page templates and configuration files, including:
	• All .opml configuration files for scheduled imports/exports
	• report-wizard.xml descriptor file for defining custom Report Wizard changes
	• componentsExtension.xml descriptor file for defining customized transactions
	• ekp.properties containing application configuration properties
	• license.ekp required to run the LMS
${base}\WEB-INF\logs	The log directory for various log files produced by the LMS.
${base}\WEB-INF\classes\com\netdimen\locale	Stores custom*.properties files that define custom labels for supported languages.

Configure the Database

This section provides instructions for creating and configuring the LMS database on supported versions of Microsoft SQL Server. For assistance with creating and configuring the LMS database on Oracle, contact PeopleFluent Customer Support.

Create the LMS Database

To create a default PeopleFluent Learning database named *EKP*, follow the steps below:

1. Log into SQL Server as a user with administrator privileges.
2. Create a new login for the LMS with user name *ndadmin* and password *ndadmin*, with SQL Server authentication selected for this login.
3. Open a new query window and enter the following to create the default LMS database and add an alias for the ndadmin user:

```
CREATE DATABASE EKP

GO

USE EKP

GO

SP_ADDALIAS ndadmin,DBO

GO
```

Deploy the Database Schema

To create the LMS database schema for Microsoft SQL Server, Oracle or MySQL, download and run the appropriate SQL script from the Database Schema section of the Release Downloads page for your LMS version:

- auto-mssql-xxxx.sql (for Microsoft SQL Server)

- auto-mysql-xxxx.sql (for MySQL)

- auto-oracle-xxxx.sql (for Oracle)

Where xxxx is the LMS version number.

Configure ekp.properties

At server startup, PeopleFluent Learning processes the ekp.properties file that defines basic system configuration information, including database connectivity details. The main purpose of this file is to provide configuration details necessary to start the LMS and set security parameters and other low-level system variables. This file is located in **<tomcat_home>\webapps\ekp\WEB-INF\conf** (where *<tomcat_home>* is the Tomcat installation folder).

Configure the following properties to connect PeopleFluent Learning to its database:

- default.connectURL

- default.user

- default.password

- default.driverName

The most commonly used properties in the ekp.properties file are listed below.

default.connectURL

This must be set to the hostname and port for the database server.

Example (Mircosoft SQL Server): **jdbc:microsoft:sqlserver:// localhost:1433;DatabaseName=ekp;SelectMethod=cursor**

default.driverName

The database JDBC driver.

Example(Mircosoft SQL Server): **com.microsoft.jdbc.sqlserver.SQLServerDriver**

default.maxconnections

The maximum number of database connections in the connection pool. This setting impacts system performance and stability. See below for more details.

Default for every database server type: **100**

default.incrementconnections

The number of new database connections allocated each time the currently allocated set is busy.

Default for every database server type: **2**

default.initialconnections

The number of database connections allocated at LMS startup.

Default for every database server type: **2**

default.retries

If no connections are available, the number of times a transaction will try to request a connection before giving up.

Default for every database server type: **5**

default.interval

The time in milliseconds to wait between each 'get connection' retry request.

Default for every database server type: **1000**

default.user

The userid used for the LMS database connection pool logins.

Default for every database server type: **ndadmin**

default.password

The password used for the LMS database connection pool logins.

Default for every database server type: **ndadmin**

system.debug

Turns on debugging information that is written to ekp.log; this should only be set to true in a test server as the overhead is considerable.

Default: **false**

system.aicctrace

Turns on AICC courseware message tracing – the messages are written to the log. This should only be set to true in a test server, as the overhead is considerable.

Default: **false**

system.smtpHost

Specifies the SMTP mail server for sending emails.

Example: **smtp.example.com**

system.allowsamelogin

Specifies same login ID can login how many times (0 = unlimited). If a value is not specified, the default is unlimited.

Default: **0**

system.forceexternalmail

Specifies whether all the messages will be sent to the external SMTP server defined in the system.smtpHost setting.

Default: **Y**

protectedPaths

Folders to protect from unauthenticated access. If unspecified, the following default values are used, overriding any setting in web.xml:

- \nd\fresco\batchreports*

- \nd\fresco\content*

- \nd\fresco\courses*

- \nd\fresco\dataloader*

- \nd\fresco\repository*

- \nd\fresco\uploads*

No default.

sessions.timeoutMinutes

Inactivity timeout period. If unspecified, a default of value of 90 will be used and will override any web.xml settings.

No default.

sessions.secure

Set to false to prohibit sessions established via HTTPS to be accessed via HTTP. If unspecified, a default value of true will be used and will override any web.xml settings.

No default.

authentication.key

Use this optional parameter to specify a secret key to access the PeopleFluent Learning API.

Default: **123456**

About default.maxconnections

The **default.maxconnections** setting can greatly impact system performance and stability. PeopleFluent recommends increasing this setting based upon the amount of RAM allocated to Tomcat's Java heap. For every 3 GB of RAM allocated, increment default.maxconnections by 50.

The relationship between Java heap RAM and default.maxconnections is shown in the table below.

RAM allocated to Java heap	Recommended default.maxconnections
3 GB	50
6 GB	100
12 GB	200
24 GB	400

Apache Solr and Zookeeper

For sites that have installed Apache Solr and Zookeeper to take advantage of scalable, external searching and indexing, you will need to add the following properties to ekp.properties:

- zookeeper.connection=ZookeeperIPAddress:2181

- serch.lucene.server=ZookeeperIPAddress:2181

Configure Apache

Why Apache

Apache acts as a proxy server between the rest of your organization's network and Tomcat.

> (i) PeopleFluent provides support only for Apache as the proxy server. While it is possible to use any web server as a proxy for Tomcat and the LMS (for example, IIS or NGinx), PeopleFluent cannot provide support for your proxy setup beyond the following general guidelines.

Example Configuration Files

PeopleFluent maintains example Apache and Tomcat configuration files in a public GitHub code repository available to customers hosting PeopleFluent Learning on-premises. Follow this link for the files:

https://github.com/PeopleFluent/learning-nts-public-resources

> (i) The example configuration files are released under the Apache License Version 2.0. Please ensure you read the LICENSE before using the configuration files.

How to Use the Configuration Files

The Apache configuration files in the GitHub repo are provided as-is and may not work for every customized environment.

The following files are provided:

- httpd.conf
- httpd-ahssl.conf
- httpd-vhosts.conf
- workers.properties
- mod_jk.conf

To use these files as-is, the following criteria must be met:

- Apache must be installed to D:\nd\Apache2
- Tomcat must be installed to D:\nd\Tomcat8
- You must create the subfolders **\jk\conf** in the D:\nd\Tomcat8\conf folder (so you have D:\nd\Tomcat8\conf\jk\conf)

Copy these configuration files to the following directories after first backing up the original files located there:

- Copy **httpd.conf** to D:\nd\Apache2\conf.

- Copy **httpd-ahssl.conf** and **httpd-vhosts.conf** to D:\nd\Apache2\conf\extra.

- Copy **workers.properties** and **mod_jk.conf** to D:\nd\Tomcat8\conf\jk\conf.

- Browse to http://archive.apache.org/dist/tomcat/tomcat-connectors/jk/binaries/windows/ and download the **mod_jk** module. For example, if you use version 1.2.40, download **tomcat-connectors-1.2.40-windows-x86_64-httpd-2.4.x.zip**.

- After unzipping the file you just downloaded, copy **mod_jk.so** to D:\nd\Apache2\modules.

Copy your SSL certificates to the D:\nd\Apache2\conf\ssl\ directory. These should include a .crt and a .key file along with the intermediary certificate authority (CA) .crt file you purchased.

Edit the following lines in httpd-ahssl.conf to include the filenames of your certificate files:

httpd-ahssl.conf
SSLCertificateFile D:\nd\Apache2\conf\ssl\netdimensions.com.no-passphrase.crt SSLCertificateKeyFile D:\nd\Apache2\conf\ssl\netdimensions.com.no-passphrase.key SSLCACertificateFile D:\nd\Apache2\conf\ssl\gd_bundle-g2-g1.crt

Finally, you need to change any domain names in the httpd-vhosts.conf and httpd-ahssl.conf files to match your domain for both ServerName values and log filename values. Search for anything named **ntstest1.netdimensions.com** and replace these values with your hostname.domain.com.

Examples of lines in httpd-vhost.conf that should be changed for your custom domain:

httpd-vhost.conf
ServerName ntstest1.netdimensions.com ErrorLog logs/ntstest1.netdimensions.com-error.log CustomLog logs/ntstest1.netdimensions.com-access.log common

Examples of lines in httpd-ahssl.conf that should be changed for your custom domain:

httpd-ahssl.conf

```
ServerName ntstest1.netdimensions.com


ErrorLog logs/ntstest1.netdimensions.com-ssl-error.log


CustomLog logs/ntstest1.netdimensions.com-ssl-access.log common


TransferLog logs/ntstest1.netdimensions.com-ssl-access.log
```

Before starting the Apache service, check the syntax of your configuration files:

1. Open a command prompt as an administrator and change directory to D:\nd\Apache2\bin.
2. Type `httpd -t` and press enter. This will test your configuration files and return *Syntax OK* if all syntax is well formed.
3. If the syntax check returns errors, review the lines in the files where these errors occur, fix them, and run the syntax check again.

After checking the configuration file syntax, start the Apache and Tomcat services and open your site in a browser (for example, mypeoplefluent.mydomain.com/ekp/login).

Install and Configure Rustici Engine

For instructions to install and configure Rustici Engine, see the Rustici Engine Installation Guide.

Appendix A – Troubleshooting

This section provides advice on troubleshooting installation problems.

Log Files

Log files are an important source of information to help troubleshoot a PeopleFluent Learning installation. The table below lists the available log files, their location on the server, and the information they contain.

Location and File Name	Description
<tomcat_home>\webapps\ekpWEB-INF\logs\ekp.log	The LMS system log file. Check this log first if the LMS does not install.
<apache_home>\webapps\logs\error.log	The Apache web server error log file.
<tomcat_home>\webapps\logs\catalina.out	This is often the best place to look if Tomcat fails to start up.
<tomcat_home>\webapps\ekp\Tools\XMLLoader\ekp-upload.log	The log file from the XMLLoader tool.
<Drive Root>:\Setup_stderr.txt	The standard error output from the installer program.

Cannot Connect to Database

If PeopleFluent Learning is unable to make a connection to the database, follow the steps below:

1. Verify that the JDBC connection URL and login information is correctly set in the ekp.properties file.
2. Verify that you have installed your license.ekp file correctly. If the license file is invalid, the LMS will show a Database connection failure error page. The ekp.log file will show the licensing error, but this is not exposed in the error page that is returned to your browser.
3. Verify that the database server is running and the database is online.
4. Log in to your database server with the credentials that you used in ekp.properties and verify that they work correctly.
5. Verify that you are able to make a network connection to your database server from the application server upon the port that your database server uses.
6. Verify that you can log in to your database server from your application server using an appropriate SQL client. Some database servers (MySQL in particular) use host-based permissions that are sensitive to the network location from which you're connecting.

ekp.log Access Errors

If the LMS cannot access the ekp.log file or it is not being updated, check its file and parent directory access permissions. The ekp.log file is located in *<tomcat_home>*\webapps\ekpWEB-INF\logs\.

Apache Error Messages

If Apache starts up but produces an error such as *Cannot find the workers file specified*, it is likely that either the TOMCAT_HOME and/or JAVA_HOME environment variables have not been properly set.

Sometimes on Windows, even when they have been set, a system reboot is required before environment variables become available to all users. Verify that the environment variables are pointing to the correct file locations, set or reset the environment variables if necessary, close Tomcat and Apache, and then restart Windows.

Tomcat Errors

When Tomcat starts from the Windows command prompt the window it opens will remains open. If the window closes immediately then an error has occurred, Tomcat has not started, and any error messages that it might have produced are not shown.

To see the error messages, you can put a *Pause* statement at the end of the *<tomcat_home>*\bin\startup.bat file. Remember to remove the pause when the error has been cleared.

The most common error when starting Tomcat is that the JAVA_HOME variable has been set incorrectly, has not been set, or cannot be read. On Windows, even if the JAVA_HOME was set correctly during installation and appears in the environment variables list, a Windows restart is required before it can be seen by other programs.

> ⓘ PeopleFluent Learning 24.04 does not support Tomcat 7. Tomcat 8 is no longer in support. PeopleFluent recommends updating to Tomcat 9.0.64 or later.

Appendix B – Performance Troubleshooting

If you experience performance issues with an on-premises upgrade or new install, with group enrollment or importing training records through the CSV data loader, there are several areas of configuration that you can update to help improve system performance.

> ⓘ These are recommendations only, and do not guarantee a performance improvement.

Tomcat Configuration

In the Tomcat configuration tool click the **Java** tab and update the initial memory pool to **6000** MB. By doing this tomcat will take a majority of the memory that it needs at the beginning instead of going back and forth to the server.

ekp.properties

In the ekp.properties file (*<tomcat_home>*\webapps\ekp\WEB-INF\conf) update the following property values:

- default.initialConnections=**0**
- default.incrementConnections=**2**
- system.allowsamelogin=[any number greater than zero] When this is set to 0, users can log in to the app as many times as they want.
- cache.learningobject=**250**
- cache.question=**250**
- cache.article=**100**
- cache.org=**250**

Add the following properties:

- sessions.timeoutMinutes=**240**
- cache.user=**100**
- cache.catalog=**100**

Secondary Connection Pool Settings

The secondary connection pool can adversely affect system performance if not configured correctly. PeopleFluent recommends updating ekp.properties with the following settings for the secondary connection pool:

- secondaryCP.maxConnections=**10**

- secondaryCP.leakDetectionThreshold=**0**

- secondaryCP.connectionTimeout=**30**

- secondaryCP.keepaliveTime=**150**

- default.maxIdleTime=**35**

- default.connectionPoolReplacementInterval=**400**

If there are still performance issues with the secondary connection pool, then add or change the following settings:

- backgroundTasks.secondaryCP=**false**

- transaction.secondaryCP=**false**

- springMvcTransaction.secondaryCP=**false**

- api.secondaryCP=**false**

Microsoft SQL Server

To fix performance issues with the TinCan API (xAPI) SQL with Microsoft SQL Server in a normal LMS schema with no replication or special considerations:

1. Stop the application (this is necessary as you will be dropping indexes and keys and stopping the application will prevent any potential data loss).
2. Back up the current database to the local drive.
3. Run the following SQL scripts in order (click the filename link to download the script file):

20200304-1-mssql-versioning_table_creation.sql

```
BEGIN TRANSACTION;

CREATE TABLE [database_script_history] (
  [execution_id] [int] IDENTITY (1, 1) NOT NULL,
  [script_name] [nvarchar](255) NOT NULL,
  [execution_time] [datetime] CONSTRAINT [DF_db_script_history_exec_time] DEFAULT
GETDATE() NOT NULL,
  [script_description] [nvarchar](1200),
  CONSTRAINT [PK_database_script_history] PRIMARY KEY ([execution_id]),
  CONSTRAINT [UK_db_scrt_history_script_name] UNIQUE NONCLUSTERED ([script_name] ASC))
GO

INSERT INTO [database_script_history] ([script_name], [script_description]) VALUES
  ('20200304-1-mssql-versioning_table_creation.sql', 'Initial creation of the script
versioning table.');

COMMIT TRANSACTION;
```

This script creates the table structure needed to track if this script was applied and can be used for any future hot fix scripts.

20200304-2-mssql-alter_tincan_statement_id_datatype.sql

```
BEGIN TRANSACTION COLUMNUPDATE;

--Create new columns for data integrity check
ALTER TABLE [tincan_statement] ADD [id_bak] char(36);
ALTER TABLE [tincan_context_activity] ADD [statement_id_bak] char(36);
ALTER TABLE [tincan_statement_action_info] ADD [statement_id_bak] char(36);
ALTER TABLE [tincan_statement_agent] ADD [statement_id_bak] char(36);
GO

IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE
TABLE_NAME='database_script_history')
  BEGIN
    ROLLBACK TRANSACTION COLUMNUPDATE;
    RAISERROR('The table [database_script_history] has not been created yet.', 16, 1);
  END
ELSE IF EXISTS (SELECT * FROM [database_script_history] WHERE [script_name] =
'20200304-2-mssql-alter_tincan_statement_id_datatype.sql')
  BEGIN
    ROLLBACK TRANSACTION COLUMNUPDATE;
    RAISERROR('This script, 20200304-2-mssql-alter_tincan_statement_id_datatype.sql,
has been executed.', 16, 1);
  END
ELSE
  BEGIN
    PRINT 'Dropping indexes and constraints...';
```

```sql
    --Drop the dependent indexes
    DROP INDEX [NUK_TINCAN_CONTEXT_STATEID] ON [tincan_context_activity];
    DROP INDEX [NUK_STATE_ACT_INFO_STATE_ID] ON [tincan_statement_action_info];
    DROP INDEX [NUK_STATE_AGENT_ID] ON [tincan_statement_agent];

    --Drop the dependent constraints and pk
    ALTER TABLE [tincan_context_activity] DROP CONSTRAINT
[FK_TINCAN_CONTEXT_STATEMENTID];
    ALTER TABLE [tincan_statement_action_info] DROP CONSTRAINT [FK_TINCAN_STATEMENTID];
    ALTER TABLE [tincan_statement_agent] DROP CONSTRAINT [FK_STATE_AGENT];
    ALTER TABLE [tincan_statement] DROP CONSTRAINT [PK_TINCAN_STATEMENT];

    --Copy the data from target columns to the new backup columns, alter the datatype
and check values
    PRINT 'Preparing to alter columns and perform data integrity check...';
    DECLARE @diffCount INT,
      @dataCheckSQL NVARCHAR(120);

    --tincan_statment
    EXEC('UPDATE [tincan_statement] SET id_bak = id');
    ALTER TABLE [tincan_statement] ALTER COLUMN [id] nchar(36) NOT NULL;

    SET @dataCheckSQL = 'SELECT @diffCount = COUNT(1) FROM tincan_statement WHERE
id_bak <> id';
    EXEC sp_executesql @dataCheckSQL, N'@diffCount INT OUTPUT', @diffCount OUTPUT

    IF @diffCount = 0
      BEGIN
        EXEC('ALTER TABLE [tincan_statement] DROP COLUMN id_bak');
      END

    --tincan_context_activity
    EXEC('UPDATE [tincan_context_activity] SET statement_id_bak = statement_id');
    ALTER TABLE [tincan_context_activity] ALTER COLUMN [statement_id] nchar(36) NOT
NULL;

    SET @dataCheckSQL = 'SELECT @diffCount = COUNT(1) FROM tincan_context_activity
WHERE statement_id_bak <> statement_id';
    EXEC sp_executesql @dataCheckSQL, N'@diffCount INT OUTPUT', @diffCount OUTPUT

    IF @diffCount = 0
      BEGIN
        EXEC('ALTER TABLE [tincan_context_activity] DROP COLUMN statement_id_bak');
      END

    --tincan_statement_action_info
    EXEC('UPDATE [tincan_statement_action_info] SET statement_id_bak = statement_id');
    ALTER TABLE [tincan_statement_action_info] ALTER COLUMN [statement_id] nchar(36)
NOT NULL;
```

```
    SET @dataCheckSQL = 'SELECT @diffCount = COUNT(1) FROM tincan_statement_action_info
WHERE statement_id_bak <> statement_id';
    EXEC sp_executesql @dataCheckSQL, N'@diffCount INT OUTPUT', @diffCount OUTPUT;

    IF @diffCount = 0
      BEGIN
        EXEC('ALTER TABLE [tincan_statement_action_info] DROP COLUMN
statement_id_bak');
      END


    --tincan_statement_agent
    EXEC('UPDATE tincan_statement_agent SET statement_id_bak = statement_id');
    ALTER TABLE [tincan_statement_agent] ALTER COLUMN [statement_id] nchar(36) NOT
NULL;

    SET @dataCheckSQL = 'SELECT @diffCount = COUNT(1) FROM tincan_statement_agent WHERE
statement_id_bak <> statement_id';
    EXEC sp_executesql @dataCheckSQL, N'@diffCount INT OUTPUT', @diffCount OUTPUT;

    IF @diffCount = 0
      BEGIN
        EXEC('ALTER TABLE [tincan_statement_agent] DROP COLUMN statement_id_bak');
      END

    PRINT 'Recreating constraints...';

    --Recreate the constraints
    ALTER TABLE [tincan_statement] ADD CONSTRAINT [PK_TINCAN_STATEMENT] PRIMARY KEY
CLUSTERED ([id] ASC) ON [PRIMARY];

    ALTER TABLE [tincan_context_activity] WITH CHECK ADD CONSTRAINT
[FK_TINCAN_CONTEXT_STATEMENTID] FOREIGN KEY ([statement_id]) REFERENCES
[tincan_statement] ([id]);
    ALTER TABLE [tincan_context_activity] CHECK CONSTRAINT
[FK_TINCAN_CONTEXT_STATEMENTID];

    ALTER TABLE [tincan_statement_action_info] WITH CHECK ADD CONSTRAINT
[FK_TINCAN_STATEMENTID] FOREIGN KEY([statement_id]) REFERENCES [tincan_statement]
([id]);
    ALTER TABLE [tincan_statement_action_info] CHECK CONSTRAINT
[FK_TINCAN_STATEMENTID];

    ALTER TABLE [tincan_statement_agent] WITH CHECK ADD CONSTRAINT [FK_STATE_AGENT]
FOREIGN KEY([statement_id]) REFERENCES [tincan_statement] ([id]);
    ALTER TABLE [tincan_statement_agent] CHECK CONSTRAINT [FK_STATE_AGENT];

    --Recreate the indexes
    CREATE NONCLUSTERED INDEX [NUK_TINCAN_CONTEXT_STATEID] ON [tincan_context_activity]
([statement_id]);
    CREATE NONCLUSTERED INDEX [NUK_STATE_ACT_INFO_STATE_ID] ON
```

```
[tincan_statement_action_info]([statement_id]);
    CREATE NONCLUSTERED INDEX [NUK_STATE_AGENT_ID] ON [tincan_statement_agent]
([statement_id]);


    --Check result to see if it can be committed or needs a rollback
    IF NOT EXISTS (SELECT *
        FROM INFORMATION_SCHEMA.COLUMNS
        WHERE ((TABLE_NAME = 'tincan_statement' AND COLUMN_NAME = 'id_bak')
          OR (TABLE_NAME = 'tincan_context_activity' AND COLUMN_NAME =
'statement_id_bak')
          OR (TABLE_NAME = 'tincan_statement_action_info' AND COLUMN_NAME =
'statement_id_bak')
          OR (TABLE_NAME = 'tincan_statement_agent' AND COLUMN_NAME =
'statement_id_bak')))
      BEGIN
        INSERT INTO [database_script_history] ([script_name], [script_description])
VALUES
          ('20200304-2-mssql-alter_tincan_statement_id_datatype.sql', 'Alter the
datatype of tincan statement ID columns and related columns to NCHAR(36).');

        COMMIT TRANSACTION COLUMNUPDATE;
        PRINT 'The operation has completed successfully. The transaction has been
committed.';
      END
    ELSE
      BEGIN
        --Debug query before rollback
        SELECT *
        FROM INFORMATION_SCHEMA.COLUMNS
        WHERE ((TABLE_NAME = 'tincan_statement' AND COLUMN_NAME = 'id_bak')
          OR (TABLE_NAME = 'tincan_context_activity' AND COLUMN_NAME =
'statement_id_bak')
          OR (TABLE_NAME = 'tincan_statement_action_info' AND COLUMN_NAME =
'statement_id_bak')
          OR (TABLE_NAME = 'tincan_statement_agent' AND COLUMN_NAME =
'statement_id_bak'));

        ROLLBACK TRANSACTION COLUMNUPDATE;
        RAISERROR('The operation has failed and transaction was rolled back. Please
check the output for details', 16, 1);
      END
  END
GO
```

This script does all the heavy lifting including dropping keys, indexes and then creates the column, moves data and then recreates those dropped objects.

4. Restart the application.

Snapshot Isolations

For Microsoft SQL Server only, run the script below to change the ALLOW_SNAPSHOT_ISOLATION and READ_COMMITTED_SNAPSHOT. This has been applied to all 15.3.0 on-premises installations and has improved performance.

EnableRCSIandSI.sql

```
/****************************************************************************
** File Name: EnableRCSIandSI.sql
** Database:
** Tracking Numbers:
** Description:Replace dbname with the correct name you are setting this option for.
** This is a database scoped option so it must be set on each db
****************************************************************************/

use master
go

SELECT name as DBBeforeChange, is_read_committed_snapshot_on,
snapshot_isolation_state_desc
FROM sys.databases
WHERE name = 'dbname';

ALTER DATABASE [dbname] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
USE [dbname];
ALTER DATABASE [dbname] SET ALLOW_SNAPSHOT_ISOLATION ON;
ALTER DATABASE [dbname] SET READ_COMMITTED_SNAPSHOT ON;
ALTER DATABASE [dbname] SET MULTI_USER;

SELECT name as DBAfterChange, is_read_committed_snapshot_on,
snapshot_isolation_state_desc
FROM sys.databases
WHERE name = 'dbname';
```

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