



PeopleFluent Learning

21.11

Upgrade Guide

Procedures for upgrading to PeopleFluent Learning 21.11

Contents

About the Upgrade Guide	3
Prerequisites	4
Upgrade Procedure	5
Summary.....	5
Provide Notice of LMS Service Suspension	6
Replace the Login Page With a Maintenance Notice Page	7
Stop the LMS Application Server.....	8
Back up the Database	9
Back up Web Application Files	10
Download the Upgrade Kit Patch	11
Prepare the Build and Restart Tomcat.....	12
Run upgrade.bat or upgrade.sh.....	14
Restore Custom Files.....	15
Apply the New License File.....	16
Start the LMS.....	17
Post-upgrade Actions	18
Appendices.....	19
Appendix A – Manual Upgrade Procedure	19
Appendix B – Upgrade Troubleshooting.....	22
Appendix C – Performance Troubleshooting.....	23
Legal Notice	30

About the Upgrade Guide

This guide explains how to upgrade PeopleFluent Learning to 21.11 for on-premises installations. Some configuration information is included for Apache, Apache Tomcat, Microsoft SQL Server, Oracle and Rustici Engine for SCORM. For more detailed installation instructions or other services not included here, please refer to vendor-specific installation documents.

If you have any issues or queries related to upgrading the LMS, 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 upgrading the LMS on-premises.
- PeopleFluent and reseller customer service staff, who may also find it 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 may 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 Revisions

Revision Information	
Revision Date:	November 5, 2021
Revision Document Version Number:	1.0
Details of Revision:	Initial publication.

Prerequisites

PeopleFluent Learning updates can only be applied sequentially when manually upgrading the LMS. You must be running version 21.08 before you can upgrade to version 21.11. Alternatively, there are upgrade kits that enable you to upgrade from 14.1, 15.1, 15.3, 21.04 or 21.08 in one step instead of incrementally. For more information, see [Download the Upgrade Kit Patch](#).

To check your current version, log into the LMS and go to: **Manage Center > System > System Activity Statistics**. The version is listed in the System Environment section.

If you are running a version earlier than 14.1, you must upgrade to 14.1 before upgrading to 21.11.

Tomcat

PeopleFluent Learning 21.11 supports a specific set of Tomcat 'point' versions for each major version:

- If your LMS server is running Tomcat 8, it must be version 8.0.42 or higher.
- If your LMS server is running Tomcat 8.5, it must be version 8.5.12 or higher.
- If your LMS server is running Tomcat 9, it must be version 9.0.37 or higher.

 PeopleFluent Learning 21.11 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 21.11 was tested exclusively using Tomcat 9.0.37. You can download Tomcat 9.0.x from <https://tomcat.apache.org/download-90.cgi>.

Throughout this document `<tomcat_home>` refers to your Tomcat installation directory.

Summary

The upgrade procedure is summarized as follows:

1. Provide users with advance notice of LMS service suspension.
2. Replace the login page with a maintenance notice page.
3. Stop the LMS application server.
4. Back up the database.
5. Back up web application files, including customized files.
6. Download the upgrade kit patch.
7. Prepare the build and restart Tomcat.
8. Run the upgrade.bat (Windows) or upgrade.sh (Unix/Linux) file (for automated update).
9. Restore custom files.
10. Copy the new license file to the webapps\ekp\WEB-INF\conf folder.
11. Start the LMS.

Notice of LMS Service Suspension

You will need to stop your current LMS application to perform the upgrade. Before the upgrade, you can use the News Manager in the LMS to provide advance notice to your users of when the LMS will be unavailable. For example, you can create an article to be pushed out to all LMS users a week or two in advance of the upgrade and set it to automatically expire after the upgrade is complete.

The news article could include:

- Expected service suspension times
- New features to be expected on upgrade
- Changes to existing features

To learn more about publishing news articles, see the following documents, which are available from the [Customer Community](#):

- System Administration Guide
- EN040 Adding News to the Login Page

During the 24-hour period before the upgrade, you may also want to use the Broadcast Messenger to send a message to all logged in users to remind them of the service suspension.

To learn more about using the Broadcast Messenger, see the System Administration Guide.

Replace the Login Page

In order to prevent users from receiving *Internal Server Error* or *404 Page not found* error pages, it is good practice to leave the web server running and to temporarily replace the login page with a maintenance notice. This notice can simply be a static HTML page stating that the system is being upgraded and when service is expected to resume. Like the default login page, it will be served by the web server without the use of the application server or Servlet engine.

Rename your usual PeopleFluent Learning login page and temporarily replace it with your Maintenance page to prevent users from logging in. The default login page for LMS is `index.html`.

Stop the LMS Application Server

To stop the PeopleFluent Learning application server, follow these steps:

1. Verify that nobody else is currently logged in by selecting **System > User Sessions** in the Manage Center.

Note that if users do not log out of the LMS but instead just close their browser window/tab, they may still be listed in the User Sessions page. Check the Last Access column to see how long the connection has been idle for; if it is longer than a day, it is likely that the user is no longer actively using the LMS.

To log out inactive users in the User Sessions screen, tick the box for each user and kill their session.

2. If there are still active users, use the Broadcast Messenger (Communications > Broadcast Messenger) to send all current users an immediate message that they should log out.
3. Stop the LMS by shutting down the application server (Servlet engine). For example, if you are using Tomcat as your Servlet engine and it is installed as a service, stop the service. Other engines (for example, WebSphere) have their own unique administration procedures for starting/stopping the server.

Back up the Database

It is essential to back up the database before updating the LMS. Please refer to your database server's documentation for detailed instructions on backing up databases.

Back up Web Application Files

Backing up the PeopleFluent Learning web application files is essential if you have a customized login page or language files.

For default installations, the website files are located in `<tomcat_home>\webapps\ekp`. This may be different if you have installed the LMS on a different application server, or with a different application name.

Copy the files to a temporary location. Do not move them—any user-uploaded files, such as skins, stylesheets and images, should always remain in their original location on the application server.

If you have changed any of the following files to customize your LMS, back them up before upgrading. You will need to reinstate any customizations that are overwritten with default files during the upgrade:

- Default login page (index.html)
- Default skins in the following folders:
 - `\nd\fresco\styles\NetD-xxx`
 - `\nd\fresco\styles\tests`
 - `\nd\fresco\images`
- Custom language files in `\WEB-INF\classes\com\netdimen\locale`
- Web application deployment descriptor file: `\WEB-INF\web.xml`

Download the Upgrade Kit Patch

For standard PeopleFluent Learning installations, you can download and run an upgrade kit executable file, which will update the LMS application and database. Alternatively, you can download the kit, extract the files you need from it and manually update the application and database by following the steps in [Appendix A – Manual Upgrade Procedure](#).

Several upgrade kits are available to upgrade to PeopleFluent Learning 21.11 that enable you to upgrade from 14.1, 15.1, 15.3, 21.04 or 21.08 in one step instead of incrementally upgrading to 21.11 one version at a time. This means that the LMS will only need to be restarted once. Upgrading multiple versions at once is also available to CFR licensed systems.

You can download the appropriate upgrade kit from the PeopleFluent Learning [Download Center](#):

1. Select **Services > Release Downloads**.
2. On the Release Downloads page select upgrade kit corresponding to the version you are upgrading from. For example, download the LMS_2111_Ugrade_Kit_patch_0.zip file if you are upgrading from either 21.04 or 21.08.

Prepare the Build and Restart Tomcat

To prepare the build

1. Unzip the upgrade kit to a folder on the application server.
2. Edit the build.xml file and set the db.username and db.password properties to the values for the account used to run database scripts during the upgrade process. These are usually the default.user and default.password values in the ekp.properties file.
3. Verify the TOMCAT_HOME and JAVA_HOME environmental variables:
 - TOMCAT_HOME must be set to your LMS web application directory.
 - JAVA_HOME must be set to the Java home directory.

In Windows, go to **Windows Start menu > Control Panel > System and Security > Advanced tab and select Environment Variables**.

In Linux, use the printenv shell command to check the environment variables:

```
printenv TOMCAT_HOME
```

```
printenv JAVA_HOME
```

Use the export (Bourne-shell) or set (C-shell) command to set the environment variables, if required. For example:

```
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_xx/jre
```

Or...

```
set TOMCAT_HOME="<tomcat_home>/webapps/ekp"
```

```
set JAVA_HOME="/usr/lib/jvm/jdk1.8.0_xx/jre"
```

4. In Windows, open the Apache Tomcat Properties either through the **Windows Start menu > Configure Tomcat**, or you can right-click the Tomcat Manager icon in the system tray.
5. In the Java tab, verify that the Java Virtual Machine reference is pointing to the jvm.dll located under \jdkx.x.x\jre\bin\server\.
6. In the Java tab, enter the amount of memory you want to allocate to the Java heap in the Maximum Memory Pool field.
7. In Linux go to the Apache Tomcat /bin directory and enter the Catalina options to set the minimum and maximum amount of RAM (MB) to allocate to the Java heap:

```
export CATALINA_OPTS="$CATALINA_OPTS -XmsMINm"
```

```
export CATALINA_OPTS="$CATALINA_OPTS -XmxMAXm"
```

Replace *MIN* with the minimum amount of RAM (MB) to allocate to the JVM heap, and replace *MAX* with the maximum amount (for example, `-Xmx12288m` sets the maximum RAM to 12 GB).

8. Restart Tomcat.

Run upgrade.bat or upgrade.sh

In Windows:

1. Change directory to the directory created when you unzipped the upgrade kit.
2. Run upgrade.bat.


In Unix or Linux:

1. Grant execution permission to ant and upgrade.sh by entering the following commands:

```
chmod +x ant
```

```
chmod +x upgrade.sh
```

2. Change directory to the directory created when you unzipped the upgrade kit.
3. Run upgrade.sh.
4. Check the error.log file in Apache's 'logs' sub-folder and contact PeopleFluent support if there are any errors.

 The upgrade kit executable file will automatically detect if the Code of Federal Regulation feature (CFR) has been enabled and will update the schema as required.

Restore Custom Files

Restore the customized files you backed up earlier in the procedure, which might include:

- Your custom login page (index.html)
- Skins in the following folders:
 - \nd\fresco\styles\NetD-xxx
 - \nd\fresco\styles\tests
 - \nd\fresco\images
- Custom language files in \WEB-INF\classes\com\netdimen\locale
- The web application deployment descriptor file: \WEB-INF\web.xml

Apply the New License File

A new license file is required only when upgrading to a major release (for example, 15.1 to 21.11). Copy the new license file to the `webapps\ekp\WEB-INF\conf` folder.

Start the LMS

Open the LMS in your browser. If the LMS does not start, [See "Appendix B – Upgrade Troubleshooting" on page 22.](#)

Post-upgrade Actions

Custom Skins

HTML structures and CSS class names can change from release to release, which may affect custom skins based on a previous release.

If your organization uses a custom skin, check that various types of page in the LMS display properly in your supported browsers. For more information about custom skins, see the Skin Customization Guide, which is available to download from the [Customer Community](#).

Custom Language Files

If you have restored custom language files to the WEB-INF\classes\com\netdimen\locale directory, check that the on-screen text displays as expected. For more information about custom language files, see the Text Customization guide, which is available to download from the Customer Community.


Apply Upgraded Language Bundles (non-US English)

Upgraded language bundles will be added to the support site as and when they become available. If your Support and Maintenance contract includes language bundles, contact PeopleFluent Customer Support to check for availability of upgraded language bundles. Availability is approximately one month after initial release and until upgraded language bundles are applied, any new or updated text will still be in US English.

If you have a PeopleFluent Learning Performance license and receive direct support from your reseller, please contact your reseller for language bundle upgrades.

Appendix A – Manual Upgrade Procedure

This appendix explains how to upgrade PeopleFluent Learning to version 21.11 manually.

 You can manually upgrade to PeopleFluent Learning 21.11 only from version 21.08.

Prepare the environment and update the web application files

1. Follow the steps for the automated upgrade procedure up to downloading the upgrade kit. After downloading the upgrade kit, unzip it to a folder on the application server.
2. If you have updated or customized any of the standard libraries, you must back them up before overwriting them with those extracted from the 21.11 upgrade kit.
3. Unzip the 2111\upgrade-ekp.zip file to the LMS web application directory (Default: `<tomcat_home>\webapps\ekp`).
4. If your LMS runs in a web context other than the default (`ekp`), ensure the `system.domain` property in `<tomcat_home>\webapps\ekp\WEB-INF\conf\ekp.properties` is set to your web context. For example, `system.domain=myWebContext`.

Prepare the LMS database

SQL Server

Before you update a SQL Server database, you must ensure the login defined in `ekp.properties` is assigned as `dbo` in the LMS database.

For example, if the default database connection properties in `ekp.properties` are as follows:

- `default.user=Undadmin`
- `default.password=Pndadmin`
- `default.driverName=com.microsoft.sqlserver.jdbc.SQLServerDriver`
- `default.connectURL=jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=ekp;SelectMethod=cursor`

Then connect to SQL Server and execute the following SQL to add the alias:

```
USE ekp
GO
sp_dropuser Undadmin
GO
sp_addalias Undadmin, dbo
GO
```

Oracle

For Oracle, the user that is used to connect to the LMS database must have read access to the *dba_objects* data dictionary table.

To grant this access in Oracle, execute the following SQL statement:

```
grant select on dba_objects to ekp;
```

Run the database upgrade script

The examples in this section use the following example database username and password:

```
default.user=Undadmin
default.password=Pndadmin
```

Note that the upgrade script can take several hours to complete, depending on the size of your LMS database.

1. Start a command prompt and change directory to `<tomcat_home>\webapps\ekp\WEB-INF\nschema\<database>`

Where `<database>` is your Oracle, SQL Server or MySQL database directory.

2. Run the command in the table below for your database.

Database	Command
Oracle	<code>sqlplus Undadmin/Pndadmin@<server> @auto-oracle-upgrade.sql</code>
SQL Server	<code>osql -Undadmin -Pndadmin -S<server> -d<database name> < auto-mssql-upgrade.sql</code>

MySQL	<code>mysql -user=Undadmin -password=Pndadmin <database name> < auto-mysql-upgrade.sql</code>
-------	--

3. If your LMS is CFR enabled, upgrade the schema for the auditing tables by entering the command in the table below for your database.

Database	Command
Oracle	<code>sqlplus Undadmin/Pndadmin@<server> @auto-oracle-audit-upgrade.sql</code>
SQL Server	<code>osql -Undadmin -Pndadmin -S<server> -d<database name> < auto-mssql-audit-upgrade.sql</code>
MySQL	<code>mysql -user=Undadmin -password=Pndadmin <database name> < auto-mysql-audit-upgrade.sql</code>

Configure the LMS

Follow these steps as needed:

1. Append the properties in the add-ekp.properties file to the end of ekp.properties in `<tomcat_home>\webapps\ekp\WEB-INF\conf\` and uncomment them as required.
2. Restore the customized files you backed up earlier in the procedure. (For more information, see [See "Restore Custom Files" on page 15.](#))
3. If you have a new license.ekp file, copy it into `webapps\ekp\WEB-INF\conf.`
4. Start the LMS and run through the post-update checks. (For more information, [See "Post-upgrade Actions" on page 18.](#))

Appendix B – Upgrade Troubleshooting

There are two types of upgrade failure:

- Upgrade kit execution failure
- Data migration failure during LMS startup

Check whether the upgrade kit completed successfully

The first thing to check is whether the upgrade kit has completed successfully. When the upgrade script runs it generates an `upgrade.log` file. If this file does not end with the message, *BUILD SUCCESSFUL*, you can find the cause of the failure in the `error.log` file.


You will need to provide PeopleFluent with the `upgrade.log` and `error.log` files for further investigation.

Check for data migration failure

When you restart your LMS Servlet engine (for example, Tomcat), it updates the `ekp.log` file. If this file contains the message, *UNABLE TO PERFORM DATA MIGRATION*, then upgrade has failed and the LMS will not start. Contact PeopleFluent Customer Support, who will need the `ekp.log` file for further investigation.

Appendix C – 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=**4**
- 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**

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_scrpt_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.

Download the script: [20200304-1-mssql-versioning_table_creation.sql](#)

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...';

    --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.

Download the script: [20200304-2-mssql-alter_tincan_statement_id_datatype.sql](#)

4. Restart the application.

Snapshot Isolations

For Microsoft SQL Server only, run the [EnableRCSIandSI.sql](#) 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

Copyright © 2021, Learning Technologies Group, Inc. All rights reserved.