**Steel Connections**

**Database Patching Mechanism**

**Contents**

[1.1 3rdparty projects database patching 4](#_Toc2592647)

[1.2 The ASProjectsVersion table 4](#_Toc2592648)

[1.3 The UpDateTable table 4](#_Toc2592649)

[1.4 The UpdateOptions table 5](#_Toc2592650)

[1.5 The patching tables 5](#_Toc2592651)

[1.6 Error logging 6](#_Toc2592652)

[1.7 Warnings 6](#_Toc2592653)

## 3rdparty projects database patching

Third party users could add their own modifications to the Steel Connections databases trough database files containing patching data. These modifications are applied during Revit start-up.  
The project patch database should be a .mdf file located in the DatabaseUpdate\Projects folder from the current database path (see DataPath value in the **ASSettings\_Advance.xml**). This folder should be created if it doesn't exist.

For Revit 2024 English US this folder could be: %ProgramData%\Autodesk\Revit Steel Connections 2024\en-US\DatabaseUpdate\Projects .

Due to the fact that Steel Connections supports multiple languages the patch database should be installed for every available language (deployed in every language folder).

The project patch database should contain the following tables:

- a project version table (**ASProjectsVersion**)

- a table describing the patch operations to be performed (**UpdateTable**)

- a table describing the patching options(**UpdateOptions**)

- tables with patching data

## The ASProjectsVersion table

The table contains information about the 3rd party project data that will be added in Steel Connections.

This table structure can be created by running the script:

CREATE TABLE [dbo].[ASProjectsVersion] (  
 [Attribute] NVARCHAR (255) NOT NULL,  
 [Value] INT NOT NULL,  
 CONSTRAINT [PK\_ASProjectsVersion] PRIMARY KEY CLUSTERED ([Attribute] ASC),  
 UNIQUE NONCLUSTERED ([Attribute] ASC)  
);

This table should contain a record with the project name and the current version number (e.g. TestProj , 1).

## The UpDateTable table

The table contains the information about the patching operations that should be run when Revit is started.

The SQL script to create this table is:

CREATE TABLE [dbo].[UpDateTable] (  
 [ID] INT NOT NULL,  
 [DSN] NVARCHAR (50) NOT NULL,  
 [SourceTable] NVARCHAR (50) NOT NULL,  
 [TargetTable] NVARCHAR (50) NOT NULL,  
 [UpdateOption] INT DEFAULT ((1)) NOT NULL,  
 CONSTRAINT [PK\_UpDateTable] PRIMARY KEY CLUSTERED ([ID] ASC),  
 UNIQUE NONCLUSTERED ([ID] ASC)  
);

The meanings of the records is:

**ID** - unique id of the patch operation  
**DSN** - the name of the Steel Connections database to be patched (this is one of the standard names specified in the **DatabaseConfiguration.xml** file located in the Configuration folder from the current database path)  
**SourceTable** - the name of the table that contains the patching information (must exists in patching database)  
**TargetTable** - the name of the Steel Connections table to be patched (in case a new table is added to the database, this is the name of the table as it will appear in Steel Connections database)  
**UpdateOption** - the type of patching operation

**UpdateOption** value should be one of the values found in the **UpdateOptions** table

## The UpdateOptions table

Script to create the table:

CREATE TABLE [dbo].[UpdateOptions] (  
 [Id] INT NOT NULL,  
 [Option] NVARCHAR (50) NULL,  
 CONSTRAINT [PK\_UpdateOptions] PRIMARY KEY CLUSTERED ([Id] ASC),  
 UNIQUE NONCLUSTERED ([Id] ASC)  
);

INSERT INTO [dbo].[UpdateOptions] ([Id], [Option]) VALUES (0, N'Merge')  
INSERT INTO [dbo].[UpdateOptions] ([Id], [Option]) VALUES (1, N'CopyTable')  
INSERT INTO [dbo].[UpdateOptions] ([Id], [Option]) VALUES (2, N'CopyTableIfNotExist')  
INSERT INTO [dbo].[UpdateOptions] ([Id], [Option]) VALUES (3, N'DeleteRows')

0 (Merge) - the rows from the source patch table are inserted in the target Steel Connections table if they don't exists, otherwise the target Steel Connections table rows are updated with the values from the source patch rows  
1 (CopyTable) - the entire source patch table is copied and eventually overwrites the target Steel Connections table  
2 (CopyTableIfNotExist) - the entire source patch table is copied in the target Steel Connections table only if this doesn't exist  
3 (DeleteRows) - the source patch table rows are deleted from the target Steel Connections table rows (matched by primary key and if there are no primary keys matched by all the columns values)

## The patching tables

These tables should have the same structure as the target Steel Connections tables and contain records to append/ modify/ delete.

The database patch is applied if the target Steel Connections database does not have the project name registered or if the current project version is lower than the patch version.  
The **TargetTable** field could contain the table name with “\*” suffix. That means the patch operation will run on all the target tables with names that begin with the string before the “\*”.

Example:

The target database could contain multiple tables with same name and different suffixes (this is the case of user profiles): **DEFAULT\_INT\_Myprofile**, **DEFAULT\_INT\_YourProfile**, etc. Specifying **DEFAULT\_INT\*** in the **TargetTable** field would apply the same patch to all these target tables.

## Error logging

If there are errors during the patch operations a log file (**PatchDatabase.log**) is created in the data path folder.

## Warnings

Please use **Merge**, **CopyTable** and **DeleteRows** options with caution as they could damage the original Steel Connections databases.

Please be careful to not delete rows that are keys in other tables and corrupt table relations.

