SIEMENS OPEN LIBRARY

7 – Library Customization

JUNE 6, 2016

Contents

1. Purpose

The purpose of this document is to provide information regarding the customization of the Siemens Open Library in order to meet applications outside of the current functionality of the Siemens Open Library. This document covers how to edit Function Blocks, User Defined Types, and Faceplates.

2. Intended Use

This document is intended to be used by anyone modifying the Open Library to customize it for applications outside of the current library functionality. This document should be used after reviewing the following and resources:

1. 1- Siemens Open Library- Library Overview and Architecture
2. 2- Siemens Open Library- Initial Setup
3. 3- Siemens Open Library- Example Object Configuration
4. 4- Siemens Open Library- Detailed Library Block Description
5. 5- Siemens Open Library- Siemens HMI Alarm Generation
6. 6- Siemens Open Library- PID Block Configuration
7. Siemens resources for creation and management of Global an Project Libraries

3. Revision History

Version	Date	Author	Comments
1.0	2016-05-23	DMC	Initial Release

4. Open Library License

Copyright (c) 2016 DMC, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

5. Reasons for Customization

This library contains a subset of all the possible features that could be required for the many different applications in order to ensure that library is easy to configure and use for the majority of applications. For this reason, it is expected that many people will make modifications to the library to fit the requirements of their industry and customers. This document is intended to assist with getting started on customization. Additionally, it is important to understand how to use a global library to help version changes and propagate them across the organization while eliminating copying and pasting of commonly used code.

6. Maintaining a New Library

The best method to capture and version changes is to utilize a Global Library. It will provide a version history and allow for easy distribution to multiple people. Edits can be made to the Siemens Open Library directly or a new Library can be created. Edits are possible as long as the library is opened with the 'read only' checkbox deselected.

7. Making Library Edits

This section will walk through the requirements needed to make edits to the Library. Any object that is going to be edited needs to be included as part of a project that contains a PLC (for UDT/Code edits) or an HMI (for Faceplate edits). If making edits with an S7-300 or S7-400, please see Section 8 for special considerations for the S7-300/400.

7.1. Order of Edits

When making edits to the Library it is best to follow this order guideline to minimize effort required to make edits.

1. Add all Global Library Objects to a project so they become part of the Project Library, as types can only be edited from the Project Library.
2. Edit the User Defined Type first, if edits are required. Since the library has versions, function blocks and faceplates will reference a specific version of the User Defined Type, so edits to Faceplates and Function Blocks will need an updated User Defined Type created first.
3. Edit the PLC Function Block, and update the type to the latest User Defined Type.
4. Edit the HMI Faceplate, and update the type to the latest User Defined Type.
5. Update the Global Library based on edits made in the Project Library.

7.2. Creating New Objects

New User Defined Types and Function Blocks can be created by first creating them on the PLC, and then dragging/dropping them into the Project Library, and then from the Project Library into the Global Library.

New Faceplates can be created by clicking 'Add New Type' in the project library, or by selecting a group of objects on the HMI, right clicking, and selecting 'Create Faceplate.'

7.3. Duplicating a Type

If intending to create a new type based on one of the existing types, then a duplicate type can be created rather than editing an existing type. This is accomplished by right clicking an object in the Project Library and selecting 'Duplicate Type.' This is now an entirely separate object and can be utilized as a starting point to build new Library Objects.

7.4. Editing User Defined Types

This section will discuss making changes to User Defined Types.

1. Add the User Defined Type into a project containing an S7-1200 or S7-1500, even if edits are being done for other processors (see special considerations for using the S7-300/400 for more information).
2. Once the User Defined Type is included in the PLC project, it will be added to the Project Library.
3. To Edit the User Defined Type, right click on the type in the Project Library and select 'Edit Type.' A pop-up will appear for the test environment. This is the environment that will be

used to verify that edits compile. For User Defined Types, verify the environment selected is an S7-1200 or S7-1500.
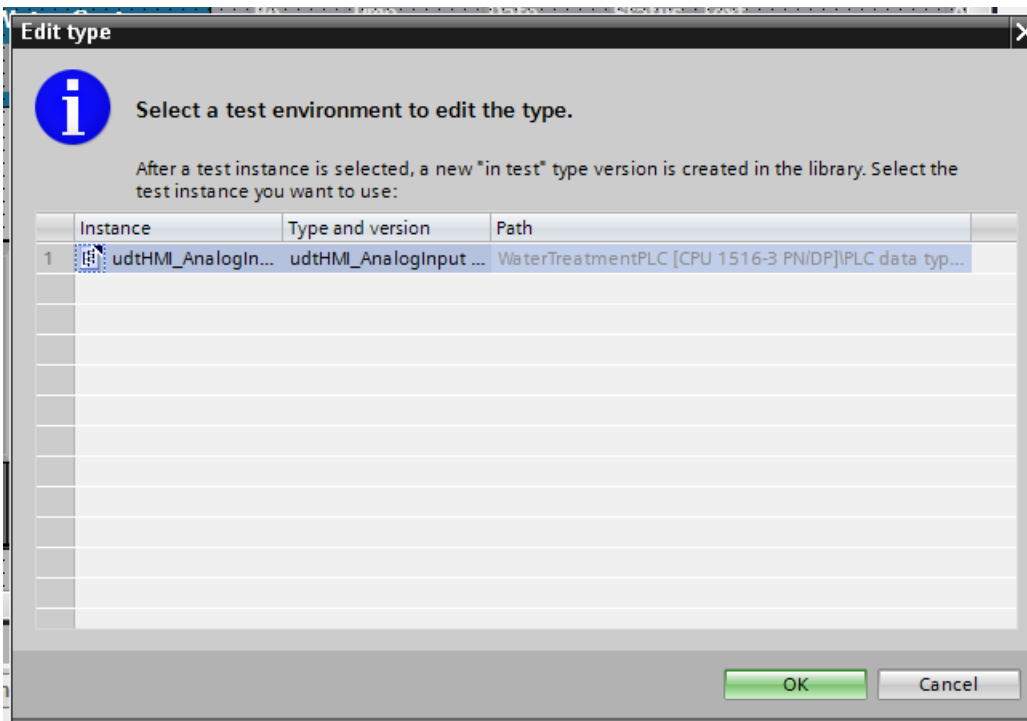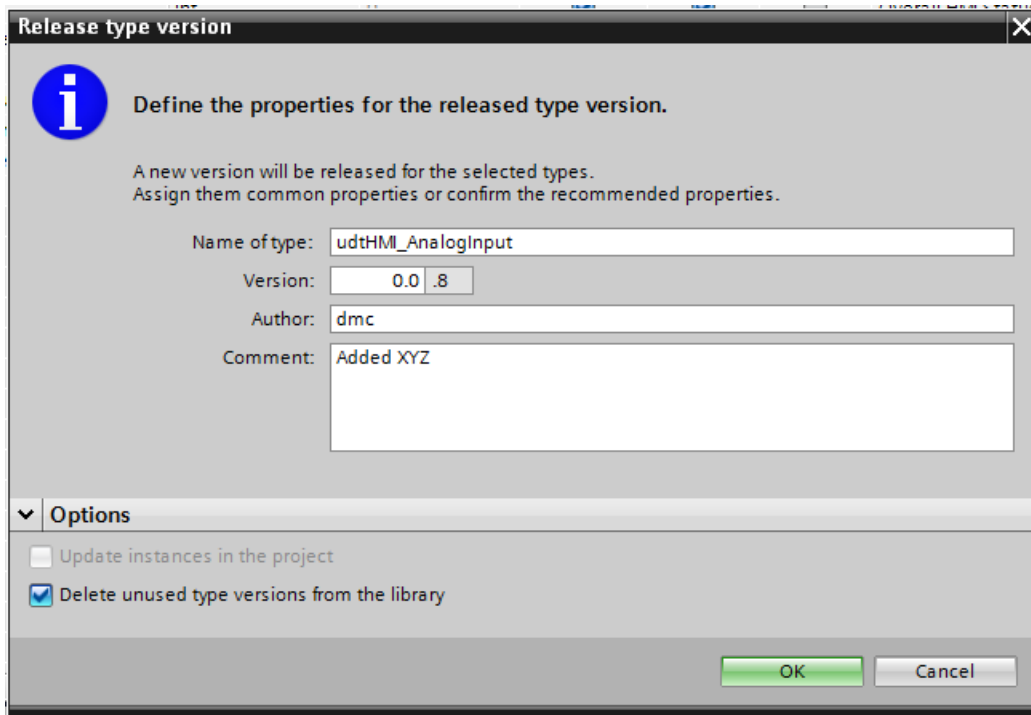

4. The editor will open, and allow you to add or remove variables as would be possible in any User Defined Type. Note that the project library shows a new version that is '[in test]'. Make any required edits to the User Defined Type and then select 'release the version.' (If the yellow banner is not visible, click the exclamation point in the yellow circle).
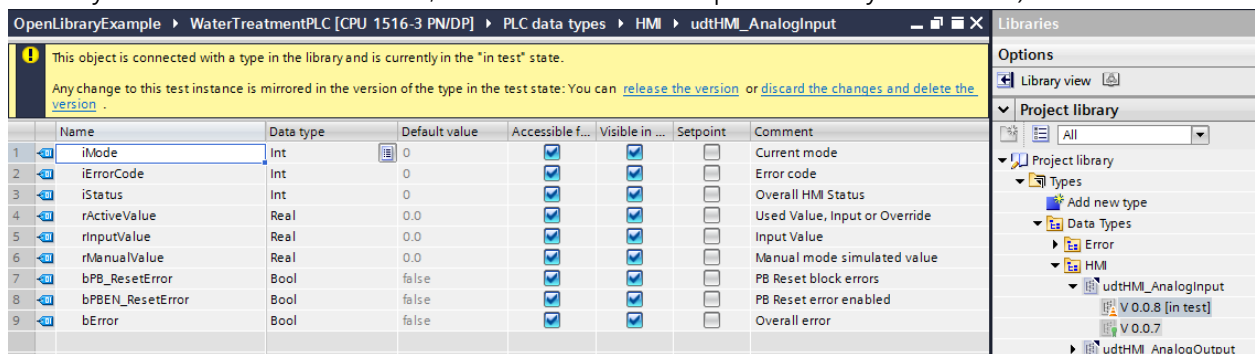
5. A pop-up will appear where version number can be selected. Additionally, it will have a check box for 'Delete unused type versions from the library.' If the Function Block and Faceplate are not part of the Project Library then it will delete the previous version of the UDT, as it won't be used in the current Project Library, however, if the Function Block and Faceplate have already been added to the project, then the previous version will remain as part of the Project Library as it is still used by the other objects in the Library. If any PLC or

HMI in the project is using the type being edited, the option 'Update instances in the project' will be available. Check this box to automatically upgrade all type instances in the project as soon as the new version is released.


6. Add the modified User Defined Type to the Global Library by dragging and dropping it into the Global Library.

7.5. Editing PLC Function Blocks
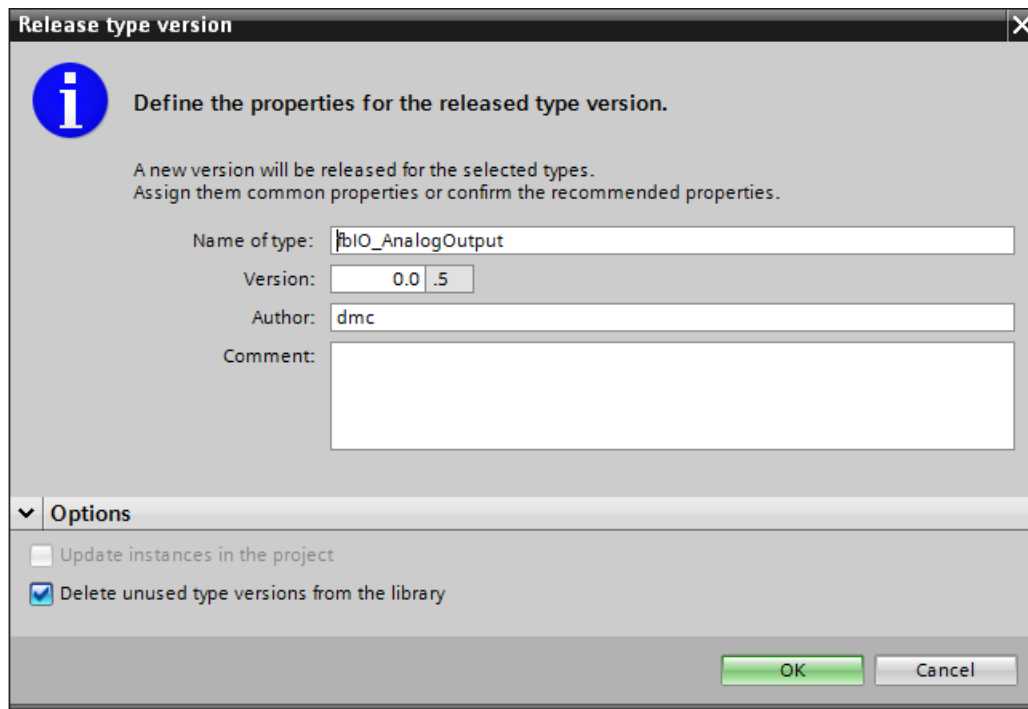This section will discuss making changes to Function Blocks.

1. Add the Function Block into a project containing a PLC that the object is to be used with (S7-300, S7-400, S7-1200, and S7-1500 have different compiler requirements).
2. If a modified User Defined Type is being used, delete the User Defined Type in the PLC Project, and add the updated User Defined Type from the Project Library.
3. To Edit the Function Block, right click on the type in the Project Library and select 'Edit Type,' or right click on the Function Block in the PLC Project and select 'Edit Type.' A pop-up will appear for the test environment. This is the environment that will be used to verify that edits compile.
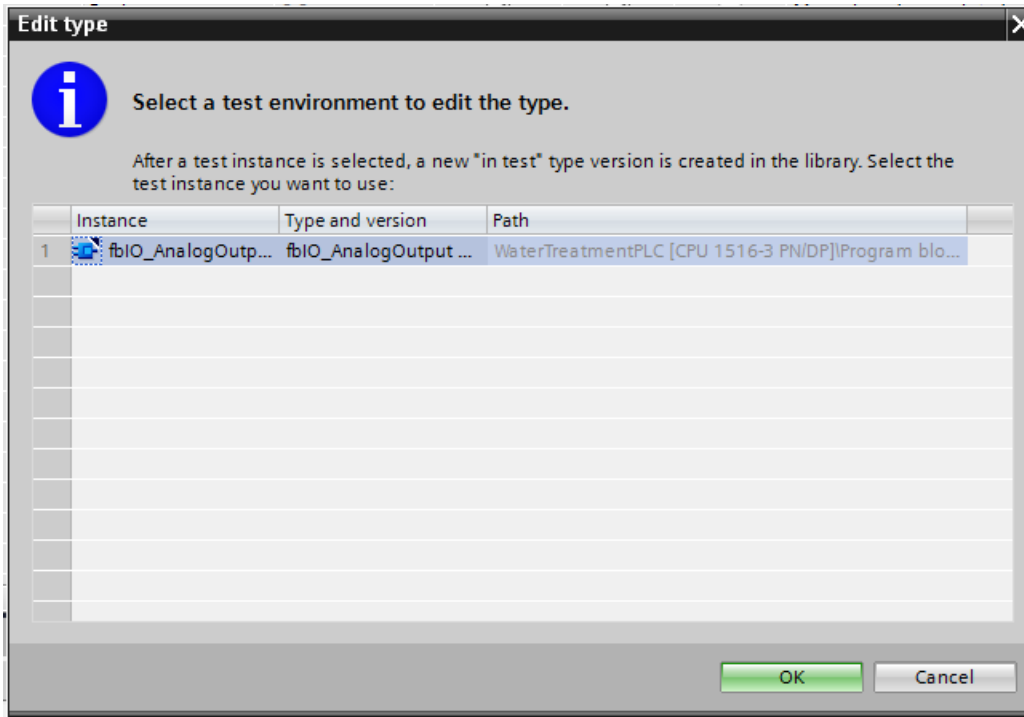
4. The editor will open, and allow you to modify code as required. Note that the project library shows a new version that is '[in test]'. Make any required edits to the Function Block and then select 'release the version.' (If the yellow banner is not visible, click the exclamation point in the yellow circle).
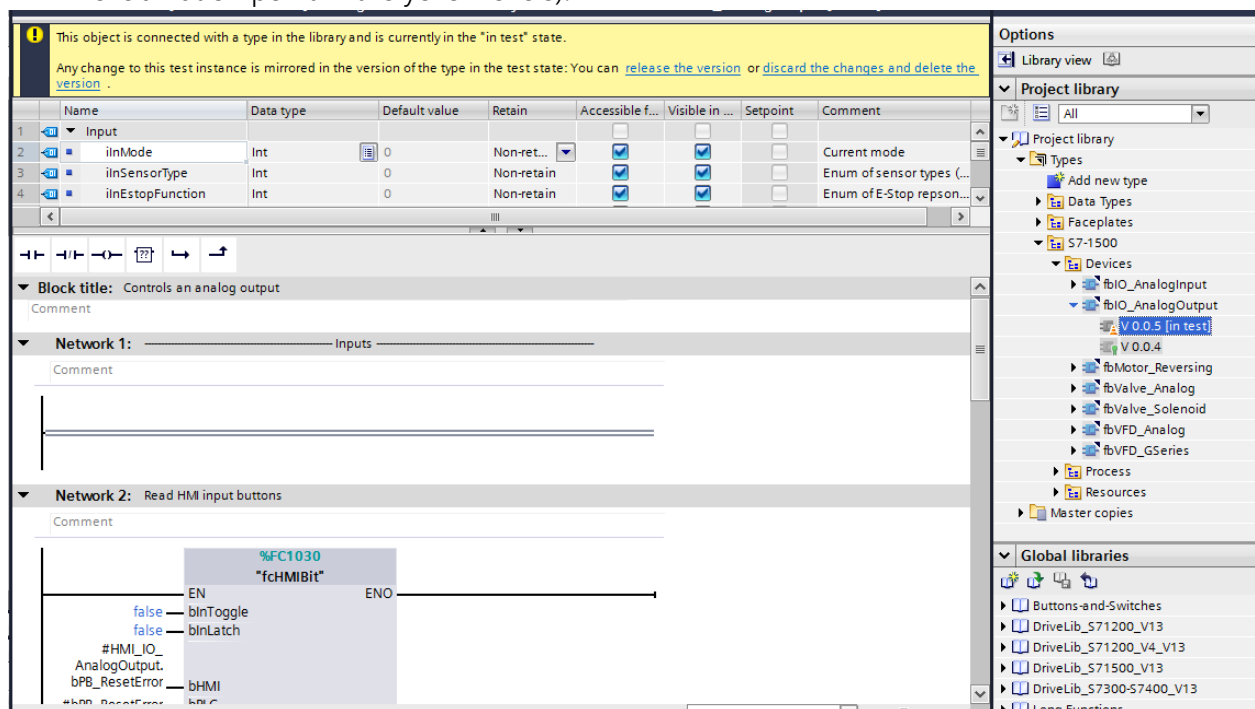
5. A pop-up will appear where version number can be selected. Additionally, it will have a check box for 'Delete unused type versions from the library,' and a checkbox to 'Update instances in the project.' If the changes made should override any previous instances of this function block, ensure that both check boxes are checked. If it is intended to keep multiple versions of the object, then ensure that neither check box is checked.


6. Add the modified Function Block to the Global Library by dragging and dropping it into the Global Library.

7.6. Editing Faceplates

This section will discuss making changes to Faceplates.

1. Add the Faceplate into a project containing an HMI (Comfort Panel or WinCC Advanced)
2. To edit the Faceplate, right click on the type in the Project Library and select 'Edit Type,' or right click on the Faceplate in the HMI Project and select 'Edit Type.'
3. The editor will open, and allow you to modify faceplate as required. Note that the project library shows a new version that is '[in test]'.
4. If a new User Defined Type is being used, select the updated version from the 'Properties' tab of the faceplate configuration.
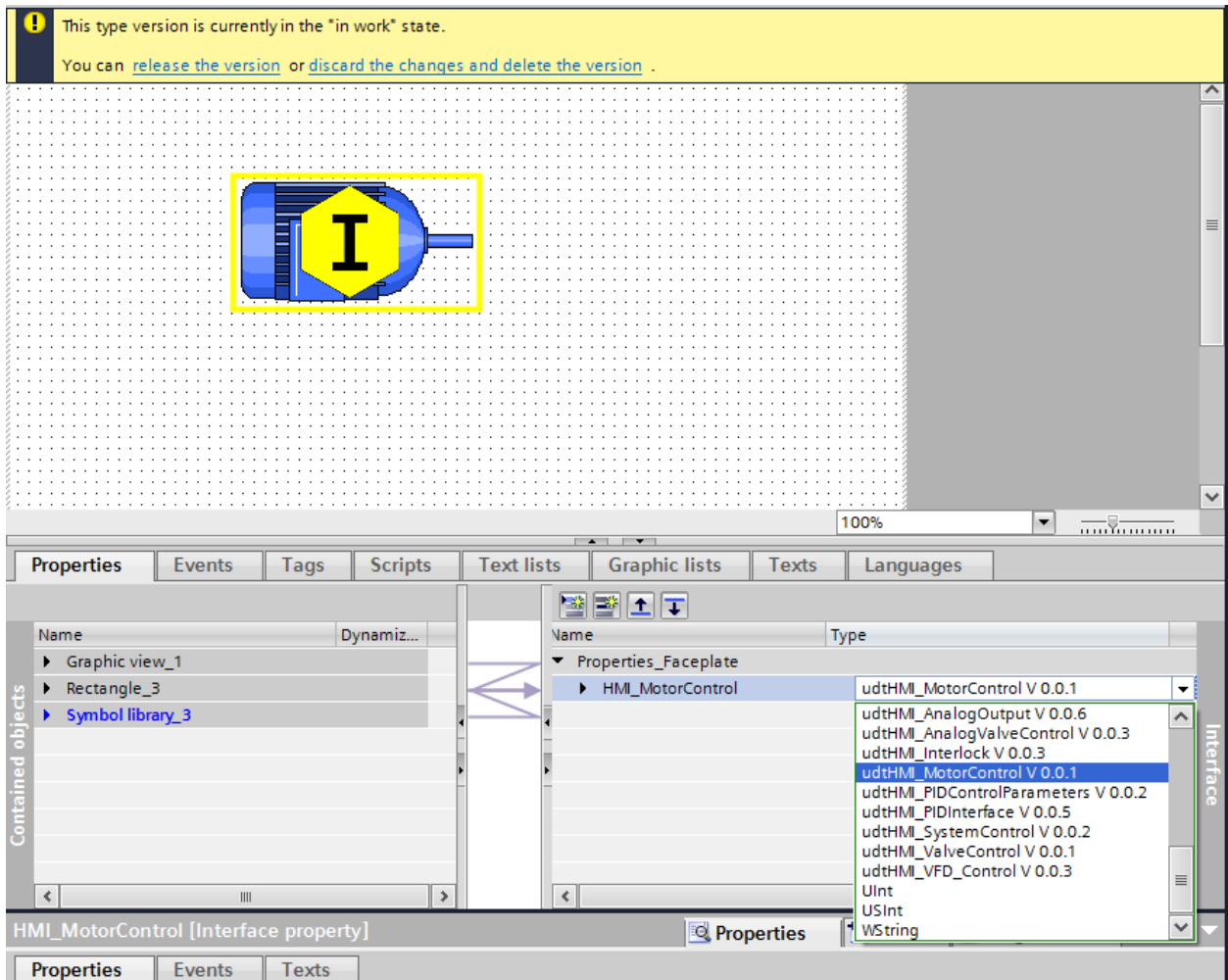
5. For objects utilizing the Siemens Object Library (VFDs, Motors, Solenoid Valve, Analog Valve, etc.) the appearance of the object can be quickly edited by selecting a different image under the 'General' portion of the properties for the object. Any changes will need to utilize 'Solid' or 'Shaded' as the Fill Style in the 'Appearance' portion of the properties for the color changing to function as intended.
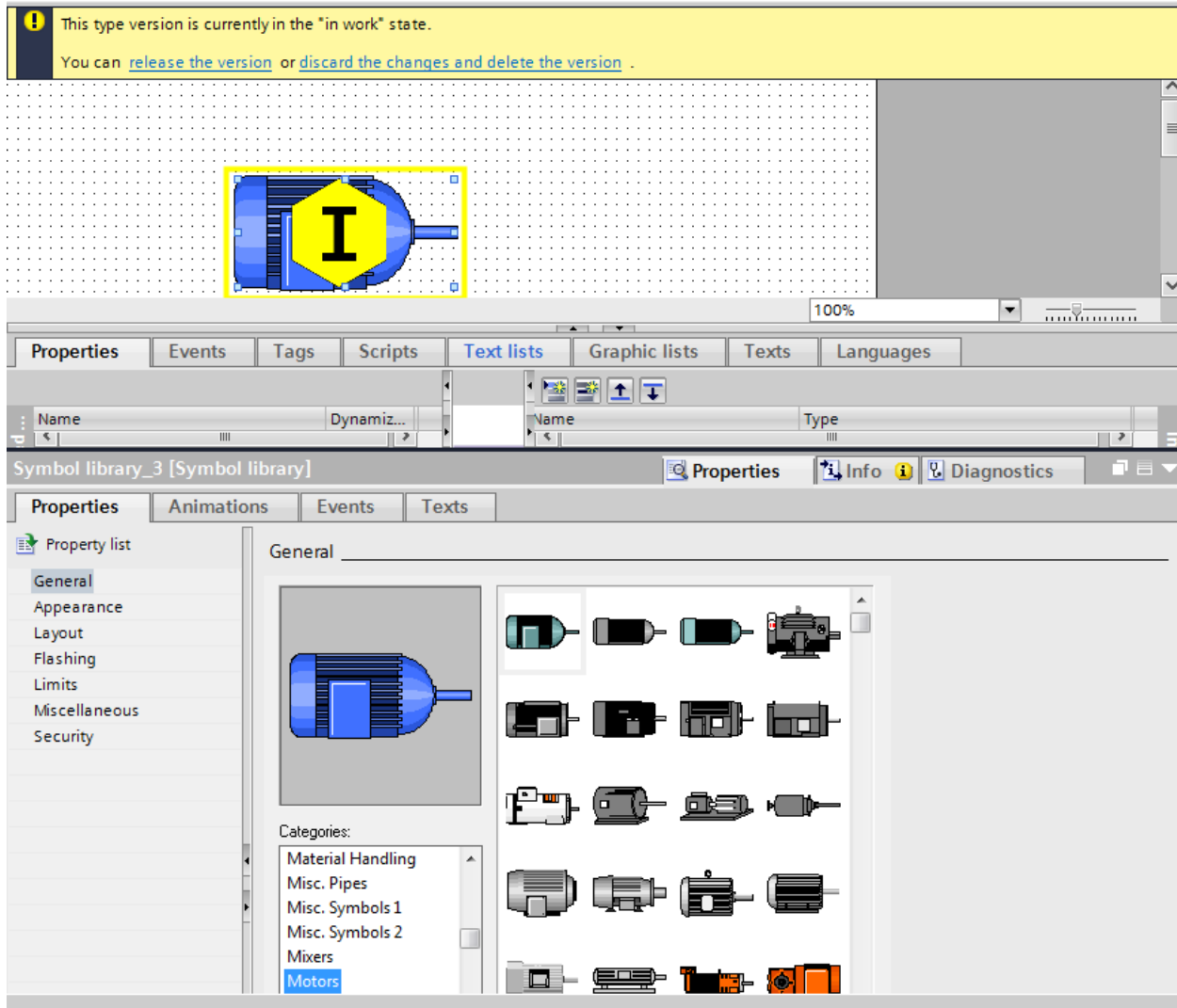
6. Make any required edits to the Faceplate and then select 'release the version.' (If the yellow banner is not visible, click the exclamation point in the yellow circle).

7. A pop-up will appear where version number can be selected. Additionally, it will have a check box for 'Delete unused type versions from the library,' and a checkbox to 'Update instances in the project.' If the changes made should override any previous instances of this function block, ensure that both check boxes are check. If it is intended to keep multiple versions of the object, then ensure that neither check box is checked.

Release type version

> **Define the properties for the released type version.**
>
> A new version will be released for the selected types.
> Assign them common properties or confirm the recommended properties.
>
> Name of type: fpMotor_Motor_Horizontal
>
> Version: 0.0 .7
>
> Author: dmc
>
> Comment:

∨ | Options

☐ Update instances in the project
☑ Delete unused type versions from the library

OK   Cancel

8. Add the modified Faceplate to the Global Library by dragging and dropping it into the Global Library.

8. Special Considerations with S7-300/400

The S7-300 and S7-400 were created before TIA Portal and TIA Portal Libraries were created, and there are limitations of User Defined Types created in the context of the S7-300 and S7-400 processors. There are also limitations when using these types in combination with a Comfort Panel or Runtime Advanced HMI. This section will cover how to make edits to the library for S7-300 and S7-400 processors without causing issues.

8.1. User Defined Types

All User Defined Types edited for the S7-300 or S7-400 should be done in the context of an S7-1200 or S7-1500. If edits are made to a User Defined Type on an S7-300 or S7-400, then those User Defined Types cannot be used with Faceplates. If edits to the User Defined Type are made on an S7-1200 or S7-1500, then that User Defined Type can be used on S7-300, S7-400, and HMI applications.

8.2. HMI User Data Type

In order for the Library faceplates to function with User Defined Types in an S7-300 or S7-400, the project library must contain an HMI User Data Type that exactly matches the structure of the PLC User Defined Type. These are included in the Library, but will need to be modified identically to match the PLC User Defined Type.

8.3. Function Block Edits

Function Block edits for the S7-300 and S7-400 should be done in the context of the PLC that they are intended to be used on. All of the processors have different functionality and compilers, and not all features are available on all devices, therefore edits should be made in the context of the PLC the code will be used on to verify that any code edits will properly compile.