©2021, openTwin

How to setup the project, use the UI and the Model, wokring with the openTwin platform in general

Create a new service

Handbook for creating a new service for the openTwin plattform

Contents

[1 Setting up the Project 2](#_Toc87109360)

[2 Understanding the classes 2](#_Toc87109361)

[2.1 Application 2](#_Toc87109362)

[2.2 dllmain 3](#_Toc87109363)

[2.3 UiComponent 3](#_Toc87109364)

[2.3.1 Create a tool menu 3](#_Toc87109365)

# 1 Setting up the Project

The easiest way to create a new service for openTwin is to use the service template provided with the openTwin plattform. The template is located at “../Libraries/ServiceTemplate”.

1. Copy the whole folder to a directory of your choice
2. Rename the folder to your desired project name
3. Rename the project files to the same name of the project folder
   1. \*.sln
   2. \*.vcxproj
   3. \*.vcxproj.filter
   4. \*.vcxproj.user
4. If the project folder contains the folder “.vs” and/or “x64”, delete them with all their contents
5. First adjust the batch files for editing and building the project:

Replace the project path to the new path the \*.vcxproj file of your new service is located at

|  |
| --- |
| "%DEVENV\_ROOT%\devenv.exe" "%SIM\_PLAT\_ROOT%\Libraries\ServiceTemplate\ServiceTemplate.vcxproj" %TYPE% "Debug|x64" /Out buildLog\_Debug.txt |

-or-

|  |
| --- |
| START "" "%DEVENV\_ROOT%\devenv.exe" "%SIM\_PLAT\_ROOT%\Libraries\ServiceTemplate\ServiceTemplate.vcxproj" |

1. Now you can open the project by running the edit.bat file.
2. In the file “Application.cpp” add the service name and service type to the ot::ApplicationBase constructor call at the top of the file. For the service type you can use one of the OpenTwin service type makros: OT\_INFO\_SERVICE\_TYPE\_<Type> or you can type in any other type.

If you are creating a service that should replace another OpenTwin service, you should use the correct service type!

1. Build the project
2. In some cases it is necessary to restart the IDE and build the project again to get all syntax highlighting right

# 2 Understanding the classes

The OpenTwin ServiceTemplate is based on the OpenTwin ServiceFoundation. The ServiceFoundation library realises all internal functionallity of a OpenTwin service.

## 2.1 Application

The central class of the Service is the Application class. This class receives all important callbacks and can be used to write down the central functionallity of the service. The virutal functions marked with override must all be implemented since the functions are pure virtual in the parent class.

## 2.2 dllmain

The dllmain.cpp file contains the startup code for the service. If modifying the file, ensure that the whole functionallity of the startup is still given.

## 2.3 UiComponent

The UiComponent is created in the service foundation. The UiComponent provides all currently implemented features for creating controls in the UI. An instance of the component will be provided whenever the UI connects or disconnects to the session, so the logic of adding controls can be implemented directly in this call.

### 2.3.1 Create a tool menu

To create a menu in the UI toolbar first a menu page must be created. In a menu page multiple groups can be added, also a group can have multiple subgroups.

After creating the page(s), group(s) and subgroup(s) controls can be added to groups or subgroups.

#### 2.3.1.1 Context menu

A button in the toolbar can get a context menu. A context menu can be created easyly by using the << operator:

|  |
| --- |
| ot::ContextMenu("WorkingPlaneContext") <<  ot::ContextMenuItem("Settings", "Settings", "Settings",  ot::ContextMenuItemRole(ot::ContextMenuItemRole::SettingsReference,  "Viewer:WorkingPlane")  ))  << ot::ContextMenuSeperator()  << ot::ContextMenuItem("Settings", "Settings", "Settings")  ); |