

# EngWrapper API

## Introduction

This API requires the version 9.1.15 or newer. When installing the software, the EngWrapper.dll will be installed inside of the binaries folder (xx-9.1). This is the file containing the EngWrapper API.

You must use a WebServer (TWebServer or IIS) configured to the software. When you create a new object for the *Project* class from the EngWrapper, it will use the WebServer to connect to the project. It is recommended to close the project before using the EngWrapper API or to open the project as a remote development. If you do not do this, you will see the new project configurations when you close and reopen the project.

Every method that adds configuration to a project adds an event to a queue. This event queue will be treated when the *Commit* method is called. The only methods that do not do this are the runtime methods and the methods that are related to the displays that start with the *IntializeCreateDisplays* and end with the *FinalizeCreateDisplays* methods.

There is a Microsoft Visual Studio solution with an example using this API available. Contact support for more information.

## EngWrapper Syntax References

### Connect to Project

#### Syntax:

```
public Project(  
  
    string projectName,  
  
    string version = "xx-9.1",  
  
    string server = "http://localhost/TProjectServer/",  
  
    bool enableLogger = false,  
  
    bool assetLevel = true)
```

#### Parameters:

- projectName = Full file name of the project
- version = Product version
- server = The TProjectServer path. Default: <http://IPAddress:Port/TProjectServer/>
- enableLogger = Flag indicating that a log is enabled
- assetLevel = Indicates that the asset level is created

#### Return:

- Project object (success)
- Error message (fail)

#### Example:

```
Project project = new Project(@"C:\Projects\Project.tproj", "xx-9.1", "http://localhost/TProjectServer/", true)
```

### Add device channel

#### Syntax:

```

public object AddDeviceChannel(
    string channelName,
    string channelProtocol,
    string channelInterface,
    string channelDescription = "",
    string channelSettings = "",
    string channelTimeout = "",
    eInitialState channelInitialState = eInitialState.Enabled,
    string channelProtocolOptions = "")

```

#### Parameters:

- channelName = Channel name
- channelProtocol = Protocol
- channelInterface = Interface type
- channelDescription = Channel description
- channelSettings = Channel settings
- channelTimeout = Timeout
- channelInitialState = Initial state
- channelProtocolOptions=Protocol options

#### Return:

- True (success)
- Exception on failure

#### Example:

```

project.AddDeviceChannel("Channell", "Modbus", "TCP/IP", "Channell", "Settings", Project.PrepareChannelTimeout
(1000, 2000, 3000, 250, 5), Project.eInitialState.Enabled,
"BlockSize=250;Encoding=RTU TCP;SingleWrite=Use single write;SlaveID=2;CoilWriteValue=FF;Offset address=false");

```

## Add device node

#### Syntax:

```

    public object AddDeviceNode(
        string nodeName,
        string nodeChannelName,
        string nodePrimaryConnection,
        string nodeSecondaryConnection = "",
        string nodeDescription = "",
        Int64 nodeCategory = 0,
        Int32 nodeLevel = 0,
        string syncSettings = "",
        string syncStation = "",
        string syncDate = "")

```

#### Parameters:

- nodeName = Node name
- nodeChannelName = Channel name
- nodePrimaryConnection = Primary station
- nodeSecondaryConnection = Secondary station
- nodeDescription = Node description
- nodeCategory = Category
- nodeLevel = Assets level

- syncSettings = Sync settings
- syncStation = Sync station
- syncDate = Sync date

**Return:**

- True (success)
- Exception on failure

**Example:**

```
project.AddDeviceNode("Node1", "Channel1", "192.168.10.19:502:1", "192.168.10.19:503:1", "Node For Channel1", 0);
```

**Add new user template****Syntax:**

```
public object AddUserTemplate(  
    string templateName)
```

**Parameters:**

- templateName = Template name

**Return:**

- True (success)
- Exception on failure

**Example:**

```
project.AddUserTemplate("PID");
```

**Add new member name in user template****Syntax:**

```
public object AddMemberName(  
    string templateName,  
    string memberName,  
    string memberComment = "",  
    object memberType = null,  
    string memberUnits = "",  
    string memberStartValue = "",  
    string memberMin = "",  
    string memberMax = "",  
    string memberArray = "",  
    string memberFormat = "",  
    string memberParameters = "",  
    eRetentive memberRetentive = eRetentive.None,  
    eDomain memberDomain = eDomain.Server,  
    eVisibility memberVisibility = eVisibility.Protected,  
    string memberDescription = null)
```

### Parameters:

- templateName = Template name
- memberName = Member name
- memberComment = Comment
- memberType = Type
- memberUnits = Units
- memberStartValue = Start value
- memberMin = Min value
- memberMax = Max value
- memberArray = Array size
- memberFormat = Format
- memberParameters = Parameters
- memberRetentive = Retentive
- memberDomain = Domain
- memberVisibility = Visibility
- memberDescription = Description

### Return:

- True (success)
- Exception on failure

### Example:

```
project.AddMemberName("PID", "sp", memberType: Project.eTagType.Double, memberDescription: "Desc SP");
```

## Add new tag name

### Syntax:

```

public object AddTag(
    string tagName,
    string tagComment = "",
    object tagType = null,
    string tagUnits = "",
    string tagStartValue = "",
    string tagMin = "",
    string tagMax = "",
    string tagArray = "",
    string tagFormat = "",
    string tagParameters = "",
    int tagLevel = 1,
    eRetentive tagRetentive = eRetentive.None,
    eDomain tagDomain = eDomain.Server,
    eVisibility tagVisibility = eVisibility.Protected,
    string tagDescription = null)

```

### Parameters:

- tagName = Tag name
- tagComment = Comment
- tagType = Type
- tagUnits = Units
- tagStartValue = Start Value
- tagMin = Min value
- tagMax = Max value
- tagArray = Array size
- tagFormat = Format
- tagParameters = Parameters
- tagLevel = Assets level
- tagRetentive = Retentive
- tagDomain = Domain
- tagVisibility = Visibility
- tagDescription = Description

### Return:

- True (success)
- Exception on failure

### Example:

```

project.AddTag("Tag1", "Comments of Tag1", Project.eTagType.Double, "V", "25", "0", "800", "", "#1",
tagDescription: "Desc Tag1");

project.AddTag("TagTemplate1", tagType: "PID", tagDescription: "Desc PID");

project.AddTag("TagEnum1", "Comments of TagEnum1", Project.eTagType.Digital, tagParameters:"Enum=Enum1;",
tagDescription: "Desc TagEnum1");

```

## Add device point

### Syntax:

```
public object AddDevicePoint(  
    string pointName,  
    string pointLabel = "",  
    string pointNode = "",  
    string pointAddress = "",  
    eDataType pointDataType = eDataType.Native,  
    string pointModifiers = "",  
    string accessTypeName = "ReadWrite",  
    string pointScaling = "")
```

#### Parameters:

- pointName = Tag name
- pointLabel = Assets level
- pointNode = Node name
- pointAddress = Address
- pointDataType = Data type
- pointModifiers = Modifiers
- accessTypeName = Access type name
- pointScaling = Scaling

#### Return:

- True (success)
- Exception on failure

#### Example:

```
project.AddDevicePoint("Tag1", "Label for Tag1", "Node1", "40001", Project.eDataType.Native, "", "ReadWrite",  
    Project.PrepareScaling(0, 65535, 0, 800));
```

## Add alarm area

#### Syntax:

```
public object AddAlarmArea(  
    string alarmArea,  
    string alarmAreaParent = "")
```

#### Parameters:

- alarmArea = Area name
- alarmAreaParent = Parent area name

#### Return:

- True (success)
- Exception on failure

#### Example:

```
project.AddAlarmArea("Level1");  
project.AddAlarmArea("Level2", "Level1");
```

## Add alarm group

### Syntax:

```
string alarmGroupName,  
eAlarmAck alarmGroupAck = eAlarmAck.Ack,  
eAlarmSound alarmGroupSound = eAlarmSound.None,  
int alarmShow = 1,  
eAlarmLog alarmLog = eAlarmLog.ActiveNormAck,  
string ackTimeout = "",  
string autoAckTimeout = "",  
string alarmGroupActiveTextColor = "",  
string alarmGroupActiveBackgroundColor = "",  
string alarmGroupAckedTextColor = "",  
string alarmGroupAckedBackgroundColor = "",  
string alarmGroupNormalTextColor = "",  
string alarmGroupNormalBackgroundColor = "",  
int activeTimeDeadband = 0)
```

### Parameters:

- alarmGroupName = Group name
- alarmGroupAck = Ack required
- alarmGroupSound = Sound
- alarmShow = Enable show
- alarmLog = Log type
- ackTimeout = Ack timeout
- autoAckTimeout = Auto ack timeout
- alarmGroupActiveTextColor = Active text color
- alarmGroupActiveBackgroundColor = Active background color
- alarmGroupAckedTextColor = Ack text color
- alarmGroupAckedBackgroundColor = Ack background color
- alarmGroupNormalTextColor = Normal text color
- alarmGroupNormalBackgroundColor = Normal background color
- activeTimeDeadband = Active time deadband

### Return:

- True (success)
- Exception on failure

### Example:

```
project.AddAlarmGroup("LogOnly", Project.eAlarmAck.Ack, Project.eAlarmSound.None, 0, Project.eAlarmLog.  
ActiveNormAck, "10000", "20000");
```

## Add alarm item

### Syntax:

```

public object AddAlarmItem(
string alarmName,
string alarmGroupName = "",
string alarmMessage = "",
Int32 alarmPriority = 0,
string alarmArea = "",
eCondition alarmCondition = eCondition.GreaterEqual,
string alarmValue = "1",
string alarmAuxValue = "",
string alarmSetPoint = "",
string alarmSetPointDeadband = "",
string alarmComment = "")

```

#### Parameters:

- alarmName = Tag name
- alarmGroupName = Alarm Group name
- alarmMessage = Message
- alarmPriority = Priority
- alarmArea = Area name
- alarmCondition = Condition
- alarmValue = Value
- alarmAuxValue = Aux Value
- alarmSetPoint = Set point
- alarmSetPointDeadband = Setpoint deadband
- alarmComment = Alarm comment

#### Return:

- True (success)
- Exception on failure

#### Example:

```

project.AddAlarmItem("Tag1", "LogOnly", "Tag1 Failure", 100, "Level2", Project.eCondition.Hi, "650");

```

## Add historical table

#### Syntax:

```

public object AddHistoricalTable(
string tableName,
int autoCreate = 1,
int saveOnChange = 1,
int timeDeadBand = 0,
int lifeTime = 0,
string trigger = "",
int saveQuality = 0 )

```

#### Parameters:

- tableName = Table name
- autoCreate = Auto create table
- saveOnChange = Save on Tag Change
- timeDeadBand = Time DeadBand
- lifeTime = File lifetime
- trigger = Save trigger



- saveQuality = Save quality

**Return:**

- True (success)
- Exception on failure

**Example:**

```
project.AddHistoricalTable("Log1", 1, 1, 1000, 10, "Tag1");
```

## Add historical item

**Syntax:**

```
public object AddHistoricalItem(  
    string pointName,  
    string pointHistoricalTable = "",  
    double deadBand = 0,  
    double deviation = 0,  
    double rateOfChange = 0)
```

**Parameters:**

- pointName = Tag name
- pointHistoricalTable = Table Name
- deadBand = Deadband
- deviation = Deviation

**Return:**

- True (success)
- Exception on failure

**Example:**

```
project.AddHistoricalItem("Tag1", "Log1", 10);
```

## Add expression

**Syntax:**

```
public object AddExpression(  
    string tagName,  
    string expression,  
    eExecution execution = eExecution.Trigger_OnChange,  
    string disableCondition = "")
```

**Parameters:**

- tagName = Tag name
- expression = Expression
- execution = Execution type
- disableCondition = Disable condition

#### Return:

- True (success)
- Exception on failure

#### Example:

```
project.AddExpression("Tag1", "Tag.Tag1=0", Project.eExecution.Trigger_OnChange);
```

## Add script class

#### Syntax:

```
public object AddScriptClass(
    string name,
    eScriptCode codeType,
    eDomain domain,
    string code)
```

#### Parameters:

- name = Script class name
- codeType = Code type
- domain = Domain
- Code = Code

#### Return:

- True (success)
- Exception on failure

#### Example:

```
string script = "public int Add(int a)" + Environment.NewLine +
    "{" + Environment.NewLine +
    "\t" + "return a + 1;" + Environment.NewLine +
    "}" + Environment.NewLine;

project.AddScriptClass("class1", Project.eScriptCode.CSharp, Project.eDomain.Server, script);
```

## Add script task

#### Syntax:

```
public object AddScriptTask(  
    string name,  
    eScriptCode codeType,  
    string trigger,  
    TimeSpan period,  
    eDomain domain,  
    string code)
```

**Parameters:**

- name = Script task name
- codeType = Code type
- trigger = Trigger
- period = Period
- domain = Domain
- Code = Code

**Return:**

- True (success)
- Exception on failure

**Example:**

```
script = "@tag2[1] = @Script.Class.Class1.Add(TK.To<int>(@tag2[1]));" + Environment.NewLine + "@tag2[2] = @tag2  
[2] + 2;" + Environment.NewLine;  
  
project.AddScriptTask("task1", Project.eScriptCode.CSharp, "Server.Second", TimeSpan.Zero, Project.eDomain.  
Server, script);
```

## Add enumeration sets

**Syntax:**

```
public object AddEnumerationSets(  
    string name,  
    string tagValue,  
    string displayValue,  
    string description = "")
```

**Parameters:**

- name = Enumeration Sets name
- tagValue = Value of tag
- displayValue = Text display value
- description = Description

**Return:**

- True (success)
- Exception on failure

**Example:**

```
project.AddEnumerationSets("Enum1", "0", "Open", "Description-Value0");  
project.AddEnumerationSets("Enum1", "1", "Close", "Description-Value1");
```

## Commits the items in the queue to the database project

### Syntax:

```
public object Commit()
```

### Return:

- True (success)
- Exception on failure

### Example:

```
project.Commit();
```

## Initialize container of displays

### Syntax:

```
public void IntializeCreateDisplays(  
System.Windows.Controls.Frame frame)
```

### Parameters:

- frame = WPF Control frame

### Return:

- None

### Example:

```
project.IntializeCreateDisplays(this.frame);
```

## Insert display

### Syntax:

```
public void InsertDisplay(  
string displayName,  
string listCSV)
```

### Parameters:

- displayName = Display name
- listCSV = String format CSV

### Return:

- None

### Example:

```
StringBuilder sb = new StringBuilder();
sb.AppendLine("Symbol,TagName,Top,Left,Labels,Uid");
sb.AppendLine("VerticalTank_Scale,,52,63,LevelValue=Tag3,VerticalTankScale3");
b.AppendLine("CurrentDisplay,,0,0,
Background=#FFFF0000;
Width=600;
Height=600;" +
            "Mode=" + Project.eDisplayMode.Popup + ";" +
            "CloseButton=1;" +
            "Title=New Title;" +
            "TitleBackground=#FFFF0000;" +
            "Border=" + Project.eDisplayBorder.Thin + ";" +
            "Placement=" + Project.eDisplayPlacement.BottomLeft + ";" +
            "Target=" + Project.eDisplayTarget.Mouse + ";" +
            "DialogButtons=" + Project.eDialogButtons.YesNoCancel + ";" +
            "Animation=" + Project.eOpenAnimation.Scroll
            );

project.InsertDisplay("Page1", sb.ToString());
```



The available display properties are: Background, Width, Height, Mode, CloseButton, Title, TitleBackground, Border, Placement, Target, DialogButtons, and Animation.

## Remove all symbols of display

### Syntax:

```
public void RemoveAllSymbols(
    string displayName)
```

### Parameters:

- displayName = Display name

### Return:

- None

### Example:

```
project.RemoveAllSymbols("MainPage");
```

## Get symbols from a display

### Syntax:

```
public string GetDisplaySymbols(  
    string displayName,  
    char sep = ',')
```

### Parameters:

- displayName = Display name
- sep = Separator

### Return:

- String format CSV

### Example:

```
string str = project.GetDisplaySymbols("MainPage");
```

## Get list of displays

### Syntax:

```
public string[] GetDisplays()
```

### Return:

- None

### Example:

```
String[] displays = project.GetDisplays();
```

## Finalize container of displays

### Syntax:

```
public void FinalizeCreateDisplays(  
    System.Windows.Controls.Frame frame)
```

**Parameters:**

- frame = WPF Control frame

**Return:**

- None

**Example:**

```
project.FinalizeCreateDisplays(this.frame);
```

**Send command to start up the runtime****Syntax:**

```
public void RunStartup(  
    string userName,  
    string password,  
    bool connect,  
    int timeoutSecWaitStarting = 60);
```

**Parameters:**

- username = User name
- password = User password
- connect = Flag to connect to runtime after starting
- timeoutSecWaitStarting = Timeout waiting runtime starts

**Return:**

- None

**Example:**

```
project.RunStartup("Guest", "", true, 60);
```

**Shutdown the runtime when started using the RunStartup method****Syntax:**

```
public void ShutdownRuntime();
```

**Return:**

- None

**Example:**

```
project.ShutdownRuntime();
```

**Check if runtime is running****Syntax:**

```
public bool IsRuntimeRunning();
```

**Return:**

- Is running (true)
- Is not running (false)

**Example:**

```
bool isRunning = project.IsRuntimeRunning();
```

**Apply changes to current runtime (Hot Start)****Syntax:**

```
public void HotStartup(string userName, string password);
```

**Parameters:**

- username = User name
- password = User password

**Return:**

- None

**Example:**

```
project.HotStartup("Guest", "");
```

**Connect to runtime to enable online configuration****Syntax:**



```
public void ConnectToRuntime(int timeoutSecWaitStarting = 60);
```

**Parameters:**

- timeoutSecWaitStarting = Timeout waiting connection to runtime

**Return:**

- None

**Example:**

```
project.ConnectToRuntime(60);
```

**Disconnect from runtime to disable online configuration****Syntax:**

```
public void DisconnectFromRuntime();
```

**Return:**

- None

**Example:**

```
project.DisconnectFromRuntime();
```

**Check if it is connected to runtime****Syntax:**

```
public bool IsConnectedToRuntime();
```

**Return:**

- Is connected (true)
- Is not connected (false)

**Example:**

```
bool isConnected = project.IsConnectedToRuntime();
```