Vijeo Designer XML Export/Import Feature

Development Kit

08/2011

EIO0000001147.00

www.schneider-electric.com

Schneider Electric

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

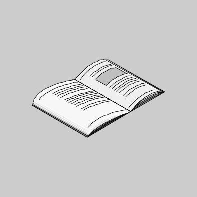
All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

Table of Contents



Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

⚡ The addition of this symbol to a Danger safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.

⚠ This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates an imminently hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a potentially hazardous situation which, if not avoided, **can result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, **can result in** minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

About the Book

At a Glance

Document Scope

This manual describes Vijeo Designer's XML Project Export/Import add-on feature that allows you to export and import project data in XML format.

Validity Note

The data and illustrations found in this book are not binding. We reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be construed as a commitment by Schneider Electric.

User Comments

We welcome your comments about this document. You can reach us by e-mail at techcomm@schneider-electric.com.

About Vijeo Designer Project Export/Import

1

Subject of this Chapter

This chapter provides a general description of Vijeo Designer's Project Export/Import feature.

What's in this Chapter?

This chapter contains the following topics:

Overview

Vijeo Designer’s XML Project Export/Import is an add-on feature that enables you to export and import project data in XML format. The main goal of this feature is to improve the interoperability of Vijeo Designer with Schneider Electric partner tools.

Once you have access to this feature, you can export Vijeo Designer project data in .vxml format, and make modifications to the project data. After modifications are made, you can import the project to Vijeo Designer.

NOTICE

RISK OF DATA LOSS

- Extreme care is necessary when you make changes to an XML project file.

Failure to follow these instructions can result in an unsuccessful import, Vijeo Designer failure, or corruption of the resulting Vijeo Designer project.

System Requirements

To access the XML Project Export/Import feature, you need the following:

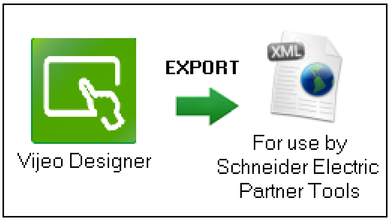
- Vijeo Designer Version 6.0 Service Pack 2 or newer
- Upgrade Reference Number.

Typical Uses of the XML Project Export/Import Feature

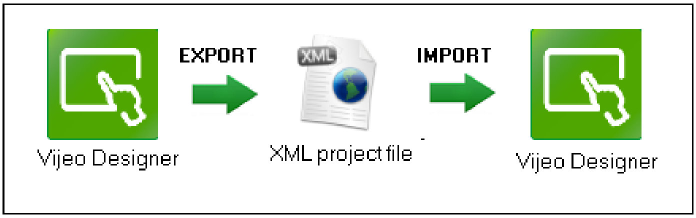
With this Vijeo Designer feature, you can export and import XML project files between Vijeo Designer and Schneider Electric partner tools for many purposes, such as backup, where you can create a text-based backup of your project.

Typical cases of XML project export and import are described as follows:

- **Case 1**: You want to export an XML project file from Vijeo Designer for use by a Schneider Electric partner tool.



- **Case 2**: You want to export an XML project from Vijeo Designer, modify the XML project file, and then import the XML project file to the same version of Vijeo Designer.



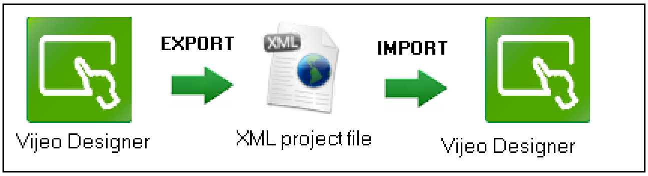
NOTICE

DATA SECURITY

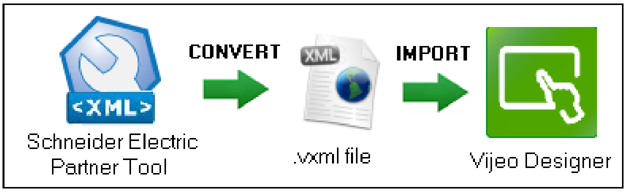
- Projects with password protection are not encrypted when exported or imported.
- Handle your data with care.
- Protect the exported XML project from unauthorized personnel.

Failure to follow these instructions can result in data tampering and unauthorized access.

- **Case 3**: You want to export an XML project file from Vijeo Designer and then import the XML project file to a newer version of Vijeo Designer.



- **Case 4**: You want to create a project using a Schneider Electric partner tool, convert the project file to an XML format that Vijeo Designer recognizes (.vxml), and then import the file to Vijeo Designer.



Working with the XML Project Export/Import Feature

2

Subject of this Chapter

This chapter describes how to use the XML Project Export/Import feature.

What's in this Chapter?

This chapter contains the following topics:

Using the XML Project Export/Import Feature

Before you can export and import XML project files, you must enter the Upgrade Reference Number that gives you access to the feature.

Entering the Upgrade Reference Number

1. In Vijeo Designer, click **Help**, and from the Help menu, select **About**.
2. In the About Vijeo Designer dialog box, click the **License** tab.
3. Click **Upgrade**.
4. Enter the Upgrade Reference Number, and click **OK**.
5. Restart Vijeo Designer.

Process to Export and Import an XML Project File

Step	Reference
1. Create a project.	*Step 1: Creating a Project*
2. Export the project data to XML format (.vxml). When exported to XML, the project data file is in Unicode Straight Text format.	*Step 2: Exporting a Vijeo Designer XML Project File*
3. In your Schneider Electric partner tool, modify the project data as required, using the schema located in the Vijeo Designer installation path.	*Step 3: Modifying XML Project Files*
4. Import the modified XML project file into Vijeo Designer.	*Step 4: Importing an XML Project File*

Step 1: Creating a Project

Using Vijeo Designer, create a project. See the Vijeo Designer online help for more information.

Step 2: Exporting a Vijeo Designer XML Project File

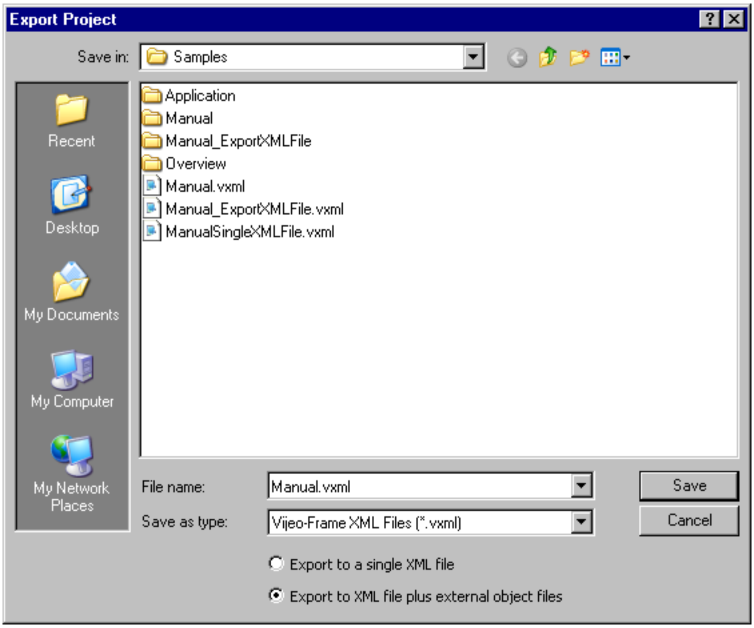
NOTICE

RISK OF DATA LOSS

- Create a backup copy of the XML project file before you make any modifications to it.

Failure to follow these instructions can result in data loss.

1. Open a Vijeo Designer project.
2. In the Navigator window's Project tab, right-click the Project node, and select **Export Project**.
3. In the Export Project dialog box, enter a .vxml project file name.



Export Project

Save in: Samples

Application
Manual
Manual_ExportXMLFile
Overview
Manual.vxml
Manual_ExportXMLFile.vxml
ManualSingleXMLFile.vxml

File name: Manual.vxml

Save as type: Vijeo-Frame XML Files (*.vxml)

○ Export to a single XML file

● Export to XML file plus external object files

4. In the **Save as type** property, select Vijeo-Frame XML Files (*.vxml).

5. Select one of the following:

   - **Export to a single XML file** - Exports a single XML file that contains all the project information. Bitmap graphic objects are text encoded and placed in the XML file.
   - **Export to XML file plus external object files** - Exports a single XML file that contains the project information and a folder that contains all the external object files (such as image files). Bitmap graphic objects are saved in native format in the extra folder.

6. Click **Save**.

   The XML project file is exported and ready for use in an XML editor.

Step 3: Modifying XML Project Files

Once exported, you can modify the XML file and then re-import it.

⚠**WARNING**

UNINTENDED EQUIPMENT OPERATION

- After importing the XML file, the changes made in the data may cause a change in equipment operation. You must test the user application thoroughly to prevent unintended equipment operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTICE

IMPORTING DATA AND PROJECTS INTO VIJEO DESIGNER

- Projects and data imported into Vijeo Designer must be checked thoroughly to make sure the data has imported successfully.

Failure to follow these instructions can result in equipment damage.

NOTICE

RISK OF DATA LOSS

- Create a backup copy of the XML project file before you make any modifications to it.
- Extreme care is necessary when you make changes to an XML project file.

Failure to follow these instructions can result in data loss.

You can modify XML files using either of these methods or a combination of both:

- Design referring to the VXML schema
- Design referring to sample exports.

Data that you cannot modify any content in the XML file is indicated with a “Do Not Modify” tag.

Design Referring to the VXML Schema

To use this method, refer to the **public.xsd** schema file for guidance. The schema file is located in the following Vijeo Designer installation path:

...\Program Files\Schneider Electric\Vijeo-Designer\Vijeo-Frame\XML

Design Referring to Sample Exports

Use this method to create an example of an object, graphical object, Toolchest parts, actions, and any other objects and settings you may want to use in the project. You then export the project to XML. In the project XML file, you can copy and paste what you created and determine the settings you want to modify.

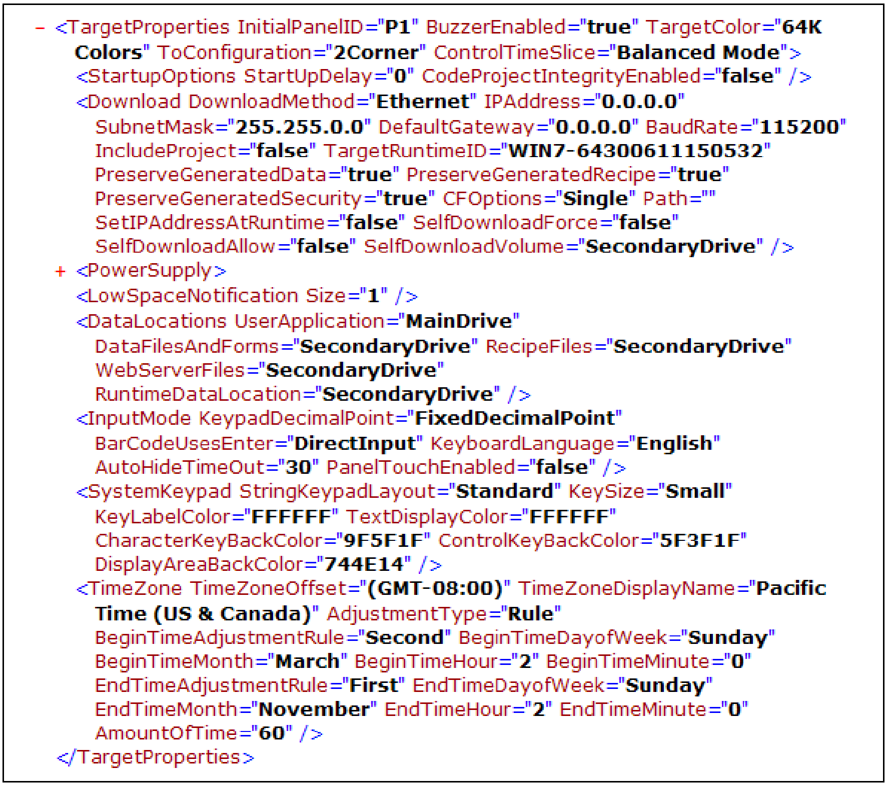
The following is an example of how you would design using sample exports.

1. In Vijeo Designer, create objects, Toolchest parts, actions, alarms, data logging, and any other objects and settings you may want to use in the project.
2. Export the project to XML from Vijeo Designer.
3. Modify the XML file. You can copy and paste the XML elements and modify the values for the attributes as needed.
4. Save the XML file.
5. Import the XML file to Vijeo Designer.
6. View your modified project.

Sample XML Data

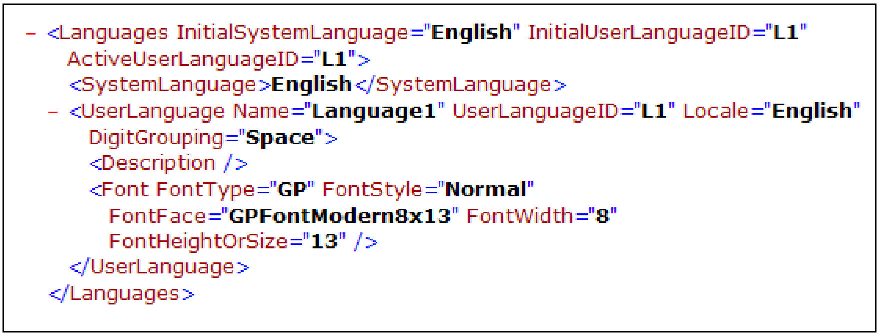
This section lists some of the data in a sample XML project file that you can add, copy, or modify. Notice that attributes for each element (such as Target Properties, Languages, Actions, Data Logging, Alarms, and so on) are enclosed within XML element tag pairs.

- Target Properties - For example, you can change the value for BuzzerEnabled from “true” to “false”.



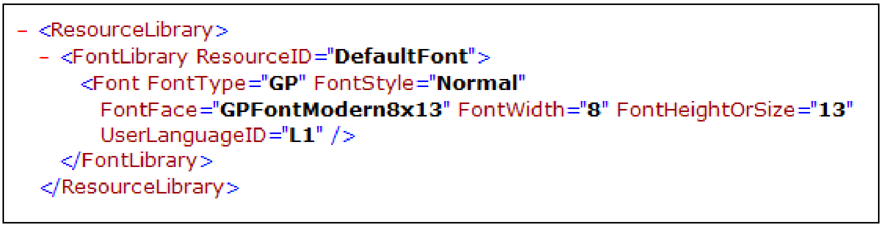
```
- <TargetProperties InitialPanelID="P1" BuzzerEnabled="true" TargetColor="64K
    Colors" ToConfiguration="2Corner" ControlTimeSlice="Balanced Mode">
    <StartupOptions StartUpDelay="0" CodeProjectIntegrityEnabled="false" />
    <Download DownloadMethod="Ethernet" IPAddress="0.0.0.0"
      SubnetMask="255.255.0.0" DefaultGateway="0.0.0.0" BaudRate="115200"
      IncludeProject="false" TargetRuntimeID="WIN7-64300611150532"
      PreserveGeneratedData="true" PreserveGeneratedRecipe="true"
      PreserveGeneratedSecurity="true" CFOptions="Single" Path=""
      SetIPAddressAtRuntime="false" SelfDownloadForce="false"
      SelfDownloadAllow="false" SelfDownloadVolume="SecondaryDrive" />
  + <PowerSupply>
    <LowSpaceNotification Size="1" />
    <DataLocations UserApplication="MainDrive"
      DataFilesAndForms="SecondaryDrive" RecipeFiles="SecondaryDrive"
      WebServerFiles="SecondaryDrive"
      RuntimeDataLocation="SecondaryDrive" />
    <InputMode KeypadDecimalPoint="FixedDecimalPoint"
      BarCodeUsesEnter="DirectInput" KeyboardLanguage="English"
      AutoHideTimeOut="30" PanelTouchEnabled="false" />
    <SystemKeypad StringKeypadLayout="Standard" KeySize="Small"
      KeyLabelColor="FFFFFF" TextDisplayColor="FFFFFF"
      CharacterKeyBackColor="9F5F1F" ControlKeyBackColor="5F3F1F"
      DisplayAreaBackColor="744E14" />
    <TimeZone TimeZoneOffset="(GMT-08:00)" TimeZoneDisplayName="Pacific
      Time (US & Canada)" AdjustmentType="Rule"
      BeginTimeAdjustmentRule="Second" BeginTimeDayofWeek="Sunday"
      BeginTimeMonth="March" BeginTimeHour="2" BeginTimeMinute="0"
      EndTimeAdjustmentRule="First" EndTimeDayofWeek="Sunday"
      EndTimeMonth="November" EndTimeHour="2" EndTimeMinute="0"
      AmountOfTime="60" />
  </TargetProperties>
```

- Languages



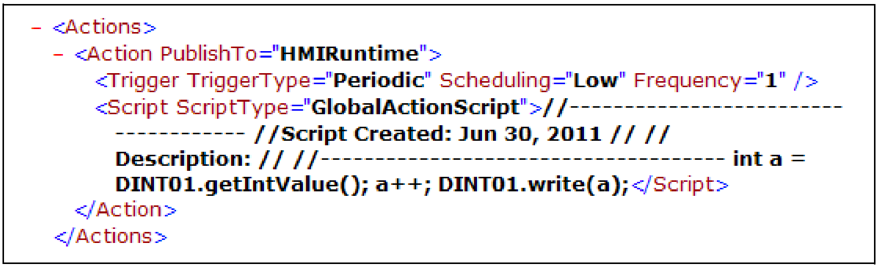
```
- <Languages InitialSystemLanguage="English" InitialUserLanguageID="L1"
    ActiveUserLanguageID="L1">
    <SystemLanguage>English</SystemLanguage>
  - <UserLanguage Name="Language1" UserLanguageID="L1" Locale="English"
      DigitGrouping="Space">
      <Description />
      <Font FontType="GP" FontStyle="Normal"
        FontFace="GPFontModern8x13" FontWidth="8"
        FontHeightOrSize="13" />
    </UserLanguage>
  </Languages>
```

- Resource Library - For example, you can change the value for FontFace from “GPFontModern8x13” to GPFont Modern13x23”. You must also update the FontWidth and FontHeightOrSize to match.



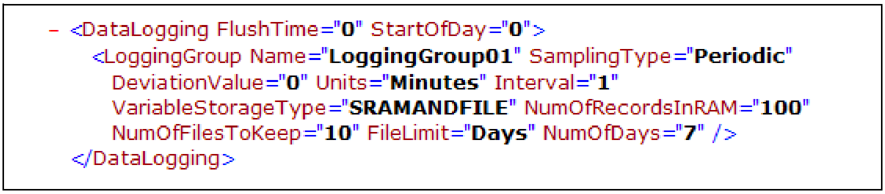
```
- <ResourceLibrary>
  - <FontLibrary ResourceID="DefaultFont">
      <Font FontType="GP" FontStyle="Normal"
        FontFace="GPFontModern8x13" FontWidth="8" FontHeightOrSize="13"
        UserLanguageID="L1" />
    </FontLibrary>
  </ResourceLibrary>
```

- Actions



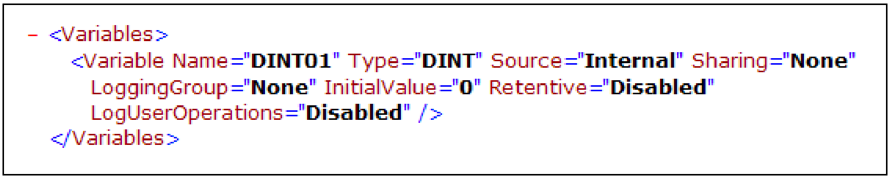
```
- <Actions>
  - <Action PublishTo="HMIRuntime">
      <Trigger TriggerType="Periodic" Scheduling="Low" Frequency="1" />
      <Script ScriptType="GlobalActionScript">//-------------------------
        ------------ //Script Created: Jun 30, 2011 // //
        Description: // //------------------------------------ int a =
        DINT01.getIntValue(); a++; DINT01.write(a);</Script>
    </Action>
  </Actions>
```

- Data Logging - For example, you can copy the DataLogging element and its attributes and paste it in your XML document, and modify the LoggingGroup Name from “LoggingGroup01” to “LoggingGroup02” The new logging group will appear but no variables would reference it. You can take a variable that is already configured for data logging and assign it to this new logging group by setting the variable’s LoggingGroup attribute to “LoggingGroup2”.



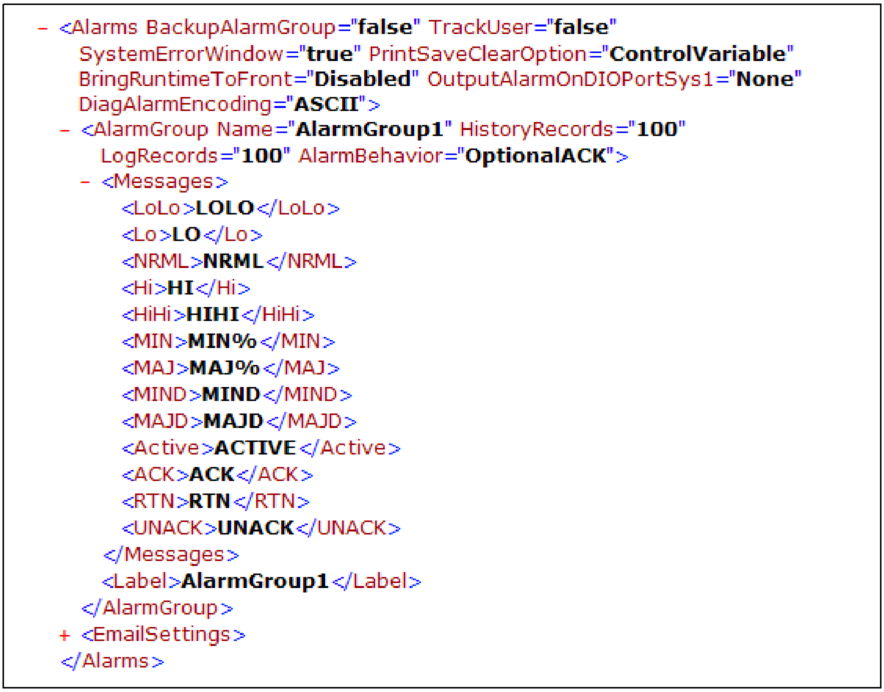
```
- <DataLogging FlushTime="0" StartOfDay="0">
    <LoggingGroup Name="LoggingGroup01" SamplingType="Periodic"
      DeviationValue="0" Units="Minutes" Interval="1"
      VariableStorageType="SRAMANDFILE" NumOfRecordsInRAM="100"
      NumOfFilesToKeep="10" FileLimit="Days" NumOfDays="7" />
  </DataLogging>
```

- Variables - For example, you can change the value in LogUserOperations from “Disabled” to “Enabled”. You must make sure when changing some attrinbutes that the objects using the variable are also updated to handle the new properties. For example, if you change the Type to “Float”, make sure the variable for your numeric display is also updated.



```
- <Variables>
    <Variable Name="DINT01" Type="DINT" Source="Internal" Sharing="None"
      LoggingGroup="None" InitialValue="0" Retentive="Disabled"
      LogUserOperations="Disabled" />
  </Variables>
```

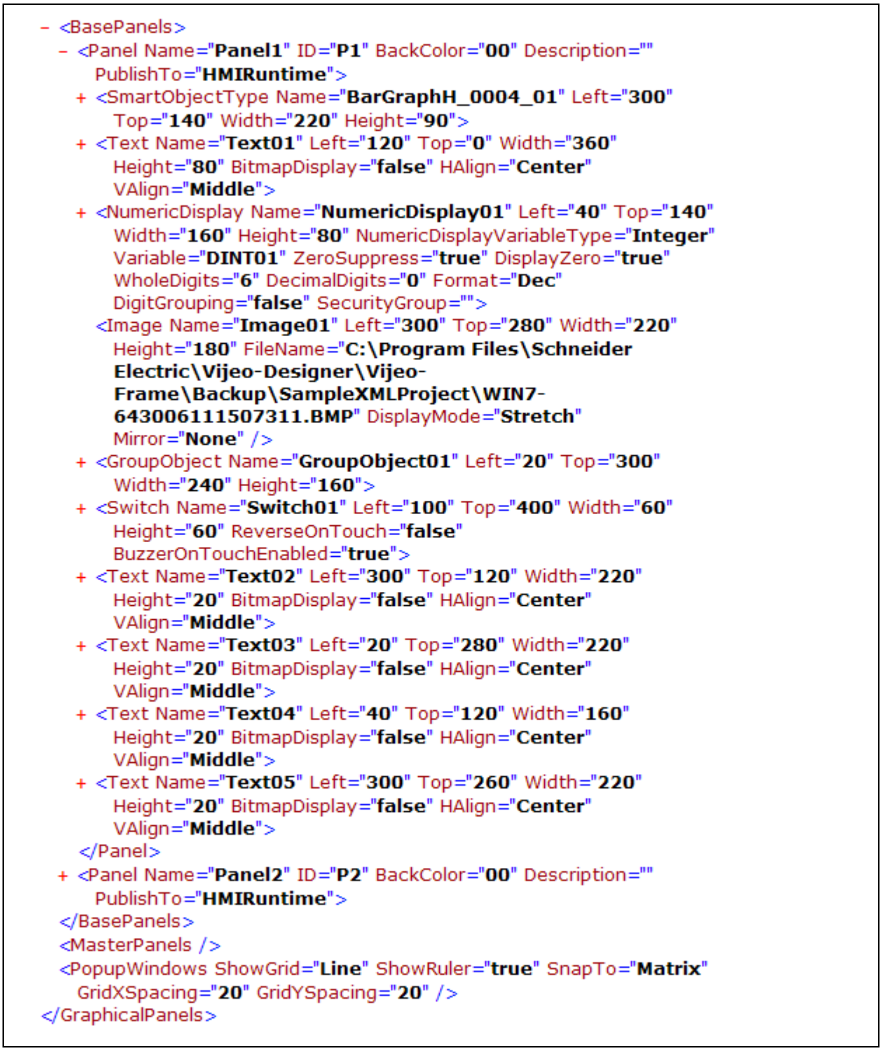
- Alarms - For example, you can change the value in HistoryRecords from “100” to “150”.



```
- <Alarms BackupAlarmGroup="false" TrackUser="false"
    SystemErrorWindow="true" PrintSaveClearOption="ControlVariable"
    BringRuntimeToFront="Disabled" OutputAlarmOnDIOPortSys1="None"
    DiagAlarmEncoding="ASCII">
  - <AlarmGroup Name="AlarmGroup1" HistoryRecords="100"
      LogRecords="100" AlarmBehavior="OptionalACK">
    - <Messages>
        <LoLo>LOLO</LoLo>
        <Lo>LO</Lo>
        <NRML>NRML</NRML>
        <Hi>HI</Hi>
        <HiHi>HIHI</HiHi>
        <MIN>MIN%</MIN>
        <MAJ>MAJ%</MAJ>
        <MIND>MIND</MIND>
        <MAJD>MAJD</MAJD>
        <Active>ACTIVE</Active>
        <ACK>ACK</ACK>
        <RTN>RTN</RTN>
        <UNACK>UNACK</UNACK>
      </Messages>
      <Label>AlarmGroup1</Label>
    </AlarmGroup>
  + <EmailSettings>
  </Alarms>
```

- Graphical Panels - For example, you can copy and paste the Switch01 element and its attributes to create Switch02. You may want to change the switch location by adding 60 to the Left property.

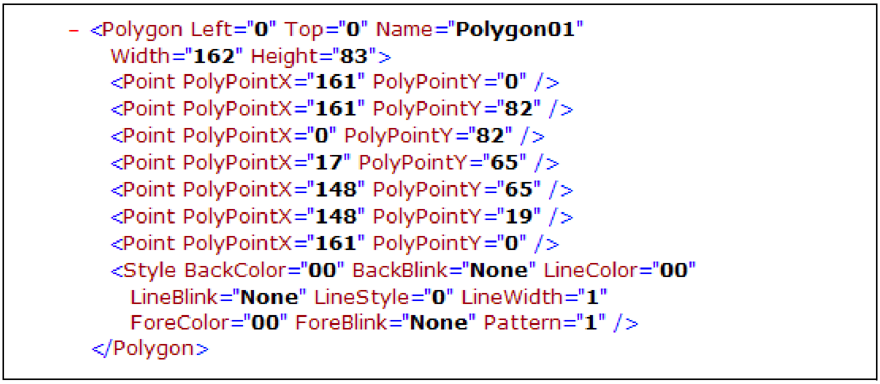
  Note that the Group Object in the below image is collapsed so you do not see the child objects within the group.



```
- <BasePanels>
  - <Panel Name="Panel1" ID="P1" BackColor="00" Description=""
      PublishTo="HMIRuntime">
    + <SmartObjectType Name="BarGraphH_0004_01" Left="300"
        Top="140" Width="220" Height="90">
    + <Text Name="Text01" Left="120" Top="0" Width="360"
        Height="80" BitmapDisplay="false" HAlign="Center"
        VAlign="Middle">
    + <NumericDisplay Name="NumericDisplay01" Left="40" Top="140"
        Width="160" Height="80" NumericDisplayVariableType="Integer"
        Variable="DINT01" ZeroSuppress="true" DisplayZero="true"
        WholeDigits="6" DecimalDigits="0" Format="Dec"
        DigitGrouping="false" SecurityGroup="">
      <Image Name="Image01" Left="300" Top="280" Width="220"
        Height="180" FileName="C:\Program Files\Schneider
        Electric\Vijeo-Designer\Vijeo-
        Frame\Backup\SampleXMLProject\WIN7-
        643006111507311.BMP" DisplayMode="Stretch"
        Mirror="None" />
    + <GroupObject Name="GroupObject01" Left="20" Top="300"
        Width="240" Height="160">
    + <Switch Name="Switch01" Left="100" Top="400" Width="60"
        Height="60" ReverseOnTouch="false"
        BuzzerOnTouchEnabled="true">
    + <Text Name="Text02" Left="300" Top="120" Width="220"
        Height="20" BitmapDisplay="false" HAlign="Center"
        VAlign="Middle">
    + <Text Name="Text03" Left="20" Top="280" Width="220"
        Height="20" BitmapDisplay="false" HAlign="Center"
        VAlign="Middle">
    + <Text Name="Text04" Left="40" Top="120" Width="160"
        Height="20" BitmapDisplay="false" HAlign="Center"
        VAlign="Middle">
    + <Text Name="Text05" Left="300" Top="260" Width="220"
        Height="20" BitmapDisplay="false" HAlign="Center"
        VAlign="Middle">
    </Panel>
  + <Panel Name="Panel2" ID="P2" BackColor="00" Description=""
      PublishTo="HMIRuntime">
  </BasePanels>
  <MasterPanels />
  <PopupWindows ShowGrid="Line" ShowRuler="true" SnapTo="Matrix"
    GridXSpacing="20" GridYSpacing="20" />
</GraphicalPanels>
```

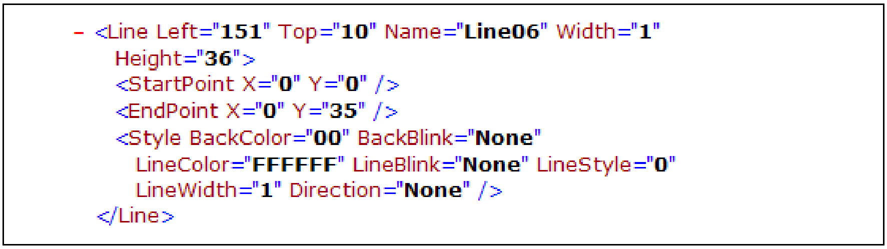
Within <GraphicalPanels> are <BasePanels>, which include images, graphical objects, and Toolchest parts – as shown in the below examples – that you can copy, paste, and modify.

- *Polygon*



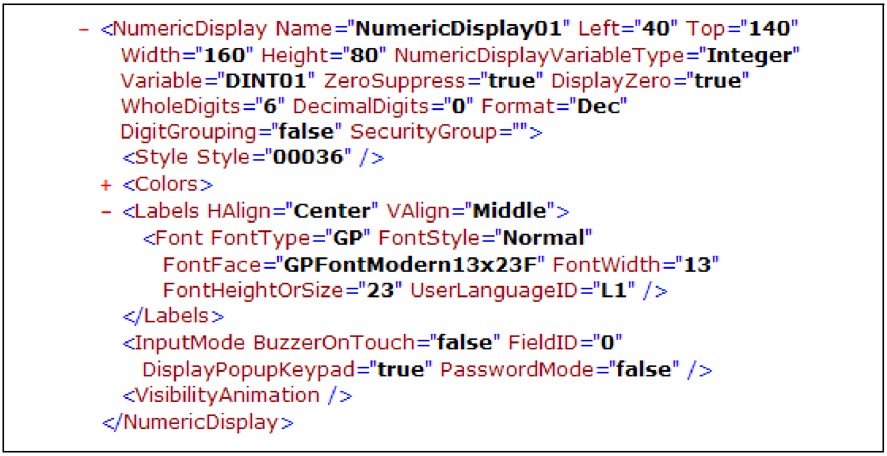
```
- <Polygon Left="0" Top="0" Name="Polygon01"
    Width="162" Height="83">
    <Point PolyPointX="161" PolyPointY="0" />
    <Point PolyPointX="161" PolyPointY="82" />
    <Point PolyPointX="0" PolyPointY="82" />
    <Point PolyPointX="17" PolyPointY="65" />
    <Point PolyPointX="148" PolyPointY="65" />
    <Point PolyPointX="148" PolyPointY="19" />
    <Point PolyPointX="161" PolyPointY="0" />
    <Style BackColor="00" BackBlink="None" LineColor="00"
      LineBlink="None" LineStyle="0" LineWidth="1"
      ForeColor="00" ForeBlink="None" Pattern="1" />
  </Polygon>
```

- *Line*



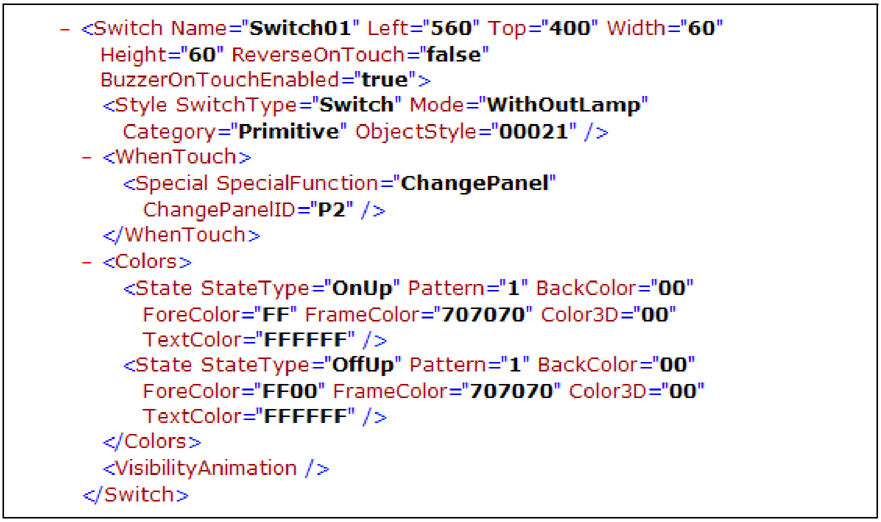
```
- <Line Left="151" Top="10" Name="Line06" Width="1"
    Height="36">
    <StartPoint X="0" Y="0" />
    <EndPoint X="0" Y="35" />
    <Style BackColor="00" BackBlink="None"
      LineColor="FFFFFF" LineBlink="None" LineStyle="0"
      LineWidth="1" Direction="None" />
  </Line>
```

- *Numeric Display*



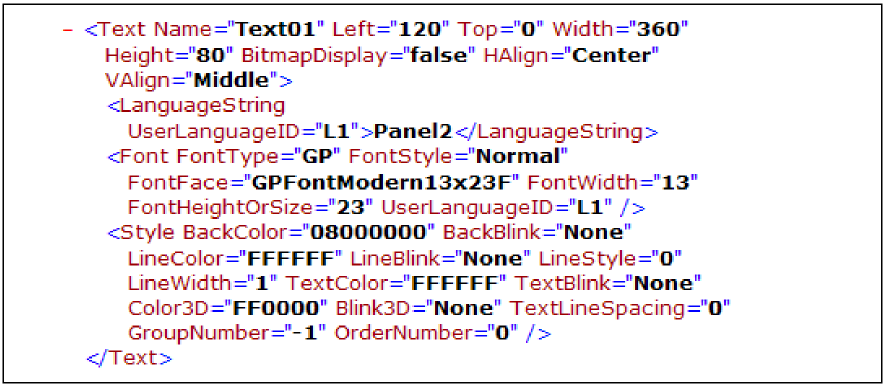
```
- <NumericDisplay Name="NumericDisplay01" Left="40" Top="140"
    Width="160" Height="80" NumericDisplayVariableType="Integer"
    Variable="DINT01" ZeroSuppress="true" DisplayZero="true"
    WholeDigits="6" DecimalDigits="0" Format="Dec"
    DigitGrouping="false" SecurityGroup="">
    <Style Style="00036" />
  + <Colors>
  - <Labels HAlign="Center" VAlign="Middle">
      <Font FontType="GP" FontStyle="Normal"
        FontFace="GPFontModern13x23F" FontWidth="13"
        FontHeightOrSize="23" UserLanguageID="L1" />
    </Labels>
    <InputMode BuzzerOnTouch="false" FieldID="0"
      DisplayPopupKeypad="true" PasswordMode="false" />
    <VisibilityAnimation />
  </NumericDisplay>
```

- *Switch*



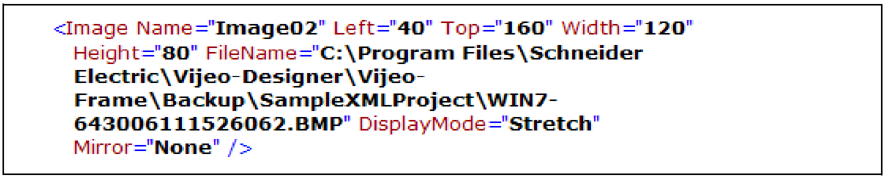
```
– <Switch Name="Switch01" Left="560" Top="400" Width="60"
    Height="60" ReverseOnTouch="false"
    BuzzerOnTouchEnabled="true">
    <Style SwitchType="Switch" Mode="WithOutLamp"
      Category="Primitive" ObjectStyle="00021" />
  – <WhenTouch>
      <Special SpecialFunction="ChangePanel"
        ChangePanelID="P2" />
    </WhenTouch>
  – <Colors>
      <State StateType="OnUp" Pattern="1" BackColor="00"
        ForeColor="FF" FrameColor="707070" Color3D="00"
        TextColor="FFFFFF" />
      <State StateType="OffUp" Pattern="1" BackColor="00"
        ForeColor="FF00" FrameColor="707070" Color3D="00"
        TextColor="FFFFFF" />
    </Colors>
    <VisibilityAnimation />
  </Switch>
```

- *Text*



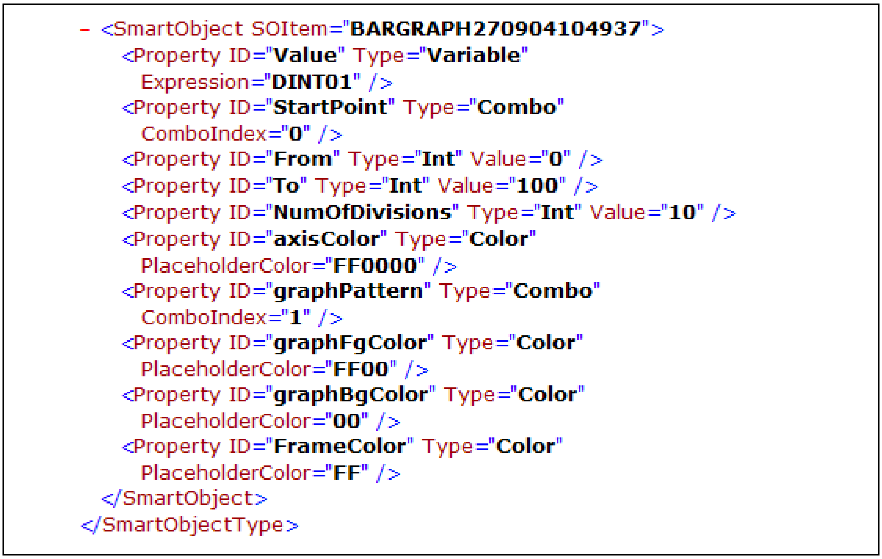
```
– <Text Name="Text01" Left="120" Top="0" Width="360"
    Height="80" BitmapDisplay="false" HAlign="Center"
    VAlign="Middle">
    <LanguageString
      UserLanguageID="L1">Panel2</LanguageString>
    <Font FontType="GP" FontStyle="Normal"
      FontFace="GPFontModern13x23F" FontWidth="13"
      FontHeightOrSize="23" UserLanguageID="L1" />
    <Style BackColor="08000000" BackBlink="None"
      LineColor="FFFFFF" LineBlink="None" LineStyle="0"
      LineWidth="1" TextColor="FFFFFF" TextBlink="None"
      Color3D="FF0000" Blink3D="None" TextLineSpacing="0"
      GroupNumber="-1" OrderNumber="0" />
  </Text>
```

- *Image*



```
<Image Name="Image02" Left="40" Top="160" Width="120"
  Height="80" FileName="C:\Program Files\Schneider
  Electric\Vijeo-Designer\Vijeo-
  Frame\Backup\SampleXMLProject\WIN7-
  643006111526062.BMP" DisplayMode="Stretch"
  Mirror="None" />
```

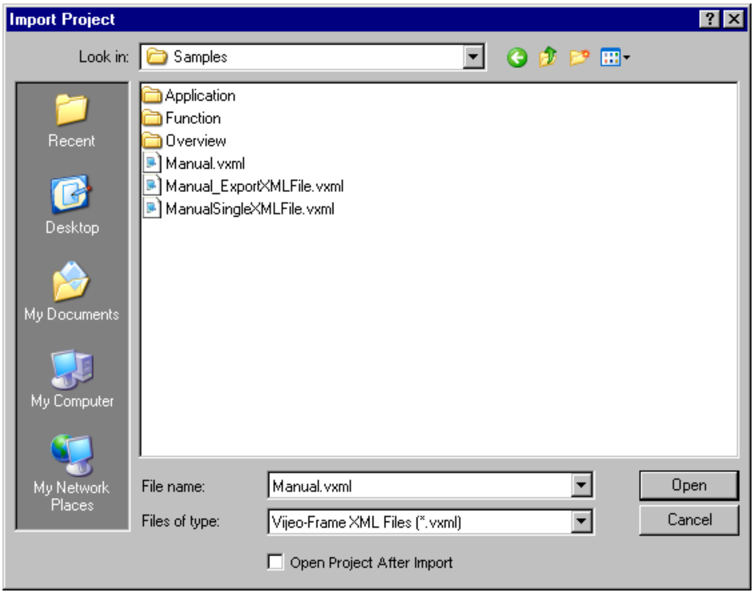
- *Bar Graph Toolchest part*



```
  - <SmartObject SOItem="BARGRAPH270904104937">
      <Property ID="Value" Type="Variable"
        Expression="DINT01" />
      <Property ID="StartPoint" Type="Combo"
        ComboIndex="0" />
      <Property ID="From" Type="Int" Value="0" />
      <Property ID="To" Type="Int" Value="100" />
      <Property ID="NumOfDivisions" Type="Int" Value="10" />
      <Property ID="axisColor" Type="Color"
        PlaceholderColor="FF0000" />
      <Property ID="graphPattern" Type="Combo"
        ComboIndex="1" />
      <Property ID="graphFgColor" Type="Color"
        PlaceholderColor="FF00" />
      <Property ID="graphBgColor" Type="Color"
        PlaceholderColor="00" />
      <Property ID="FrameColor" Type="Color"
        PlaceholderColor="FF" />
    </SmartObject>
  </SmartObjectType>
```

Step 4: Importing an XML Project File

1. In the Navigator window's Vijeo-Manager tab, right-click the **Projects** node, and select **Import Project**.
2. In the Import Project dialog box, select the .xvml file.



Import Project

Look in: Samples

Application
Function
Overview
Manual.vxml
Manual_ExportXMLFile.vxml
ManualSingleXMLFile.vxml

File name: Manual.vxml

Files of type: Vijeo-Frame XML Files (*.vxml)

☐ Open Project After Import

3. Click **Open**.

   Vijeo Designer validates the XML file against the schema.

   If the XML project file completes the validation successfully, the file is imported and ready for use in Vijeo Designer. The imported file retains its original project name, regardless of the file name. If a project already exists with the same name, the project is overwritten by the imported file.

   If the XML project file does not pass a limited validation based on the schema, a list of errors display in a dialog box. You must open your XML project file and fix the file contents adhering to the **public.xsd** schema file in the following Vijeo Designer installation path:

   ...\Program Files\Schneider Electric\Vijeo-Designer\Vijeo-Frame\XML