UNICOS: UNIFIED INDUSTRIAL COCONTROL SYSTEM

CPC (CONTINUOUS PROCESS CONTROL)

BASIC COURSE

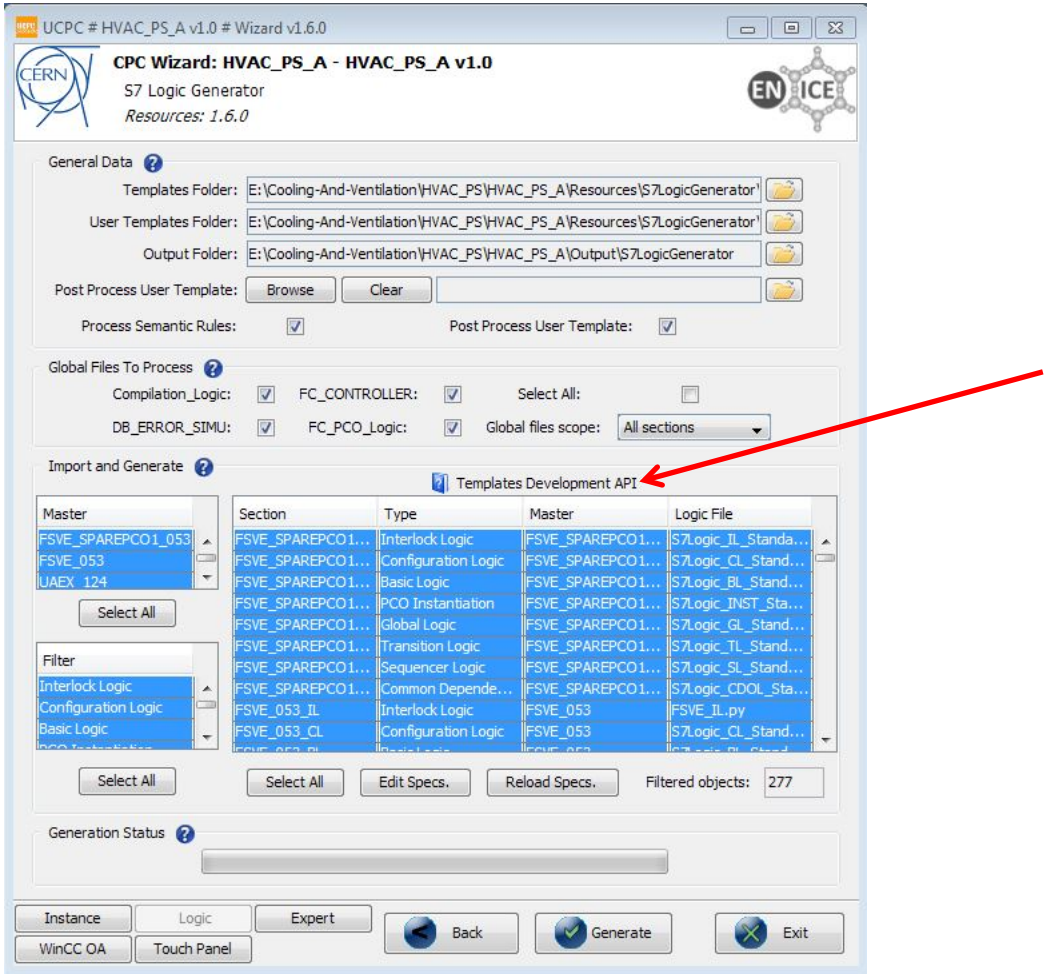
SESSION 3: PLC LOGIC TEMPLATES

JYTHON FUNCTIONS

CERN EN/ICE group

DOCUMENTATION

Click on the Logic Generator info link



Main jython Interfaces

IS7SymbolTemplate. Interface to provide methods to access the Step 7 symbols.

ISCADAPlugin. Interface to provide methods to access the SCADA plug-in.

IGenerationPluginTemplate. This interface provides the common methods used in the Jython templates to interact with the generation plug-ins.

ILogWriterTemplate. Interface containing the methods that can be used from the Jython templates to write in the user report window.

- **IDeviceInstanceTemplate.** Interface used to handle the device type instances available in the specifications file. The methods described in this interface can be used by the Jython template developers to complete the logic templates.

- **IDeviceTypeTemplate.** Interface used to handle the device types available in the specifications file. The methods described in this interface can be used by the Jython template developers to complete the logic templates.

ISpecDocumentation. Interface to read/write from the Specs' Project Documentation sheet.

- **ISpecFileTemplate.** Interface used to get data from the specifications file. The methods described in this interface can be used by the Jython template developers to complete the logic templates.

INSTANCES INTERFACE METHODS

- **getDeviceType** (IdeviceType) Method used to get the device type of the current instance. Returns the device type of the current instance.

- **getDeviceTypeName** (string) Method used to get the device type name of the current instance. Returns the device type name of the current instance.

- **getInstanceNumber** (int) Get the number of the current instance as defined in the specifications file. Returns the instance number as defined in the specifications file.

- **getAttributeData.** (string) Returns the data associated with the instance attribute as defined in the specifications file

Example: Display the name of all the Digital Input instances.

```
# Get the DigitalInput device type
diDeviceType = theRawInstances.getDeviceType("DigitalInput")
# Get a vector with all the DigitalInput instances
diInstances = diDeviceType.getAllDeviceTypeInstances()
# Display the name of all the DigitalInput instances in the UAB log file (as a debug message)for
instance in diInstances:
thePlugin.writeDebugInUABLog(instance.getAttributeData("DeviceIdentification:Name”)
```

DEVICE TYPES INTERFACE METHODS

- **getDeviceTypeName** (string) Method used to get the device type name. Returns a String containing the device type name.

- **getObjectType** (string) Method used to get the ObjectTypeFamily from the device type definition. Returns a String containing the family type of the device type. Currently, the available families are:
  - IOObjectFamily
  - InterfaceObjectFamily
  - FieldObjectFamily
  - ControlObjectFamily

- **getSpecificationAttributes** (list) Get the list of specification attributes (e.g.: DeviceIdentification:Name, DeviceIdentification:Expert Name, ...) Returns the list of specification attributes.

- **getDescription** (string) Get the device type description as specified in the device type sheet. Returns a String containing the device type description if exists, otherwise an empty string.

DEVICE TYPES INTERFACE METHODS

- **getAllDeviceTypeInstances** (vector) Method used to get a vector containing all the instances of the device type. Returns a vector containing all the instances of the device type.

Example: Display the name of all the DigitalInput instances.

```
# Get the DigitalInput device typediDeviceType = theRawInstances.getDeviceType("DigitalInput")
# Get a vector with all the DigitalInput instances
diInstances = diDeviceType.getAllDeviceTypeInstances()
# Display the name of all the DigitalInput instances in the UAB log file (as a debug message)
for instance in diInstances:
    thePlugin.writeDebugInUABLog(instance.getAttributeData("DeviceIdentification:Name"))
```

- **getDeviceTypeInstance** (IdeviceInstance) Method used to get a device type instance by its instance number. Returns the requested instance object as specified by the user if it exists, otherwise null.

Example: Get the instance number 5 of the DigitalInput device type.

```
# Get the DigitalInput device type
diDeviceType = theRawInstances.getDeviceType("DigitalInput")
# Get the DigitalInput instance number 5
diInstance = diDeviceType.getDeviceTypeInstance(5)
# Display the name of the DigitalInput instance in the UAB log file (as a debug message)
thePlugin.writeDebugInUABLog(diInstance.getAttributeData("DeviceIdentification:Name"))
```

SPECS INTERFACES METHODS

Method Summary	
String	**createSectionText**(Vector<IDeviceInstance> theTypeInstances, int optionDuplicate, int counterStart, int counterStep, String textIfLink, String textIfNoLink) This method is used to generate text for a program/section.
String	**createSectionText**(Vector<IDeviceInstance> theTypeInstances, int counterStart, int counterStep, String textRepeated) This method is used to generate text for a program/section.
int	**Dcount**(String field, String type, String condition) Count the number of elements found by the Dlookup(String, String, String) method. Iterates through all the instances of the device type specified.
Vector<String>	**DependentLoop**(String deviceTypeName, String master, int start, int step, String textRepeated) Create a String vector with the 'textRepeated' for each device instance matching the specified conditions.
Vector<String>	**DependentLoop**(String deviceTypeName, String master, int start, int step, String textRepeated, String condition) Create a String vector with the 'textRepeated' for each device instance matching the specified conditions.
String	**DependentLoopString**(String deviceTypeName, String master, int start, int step, String textRepeated) Similar to DependentLoop(String, String, int, int, String).
String	**DependentLoopString**(String deviceTypeName, String master, int start, int step, String textRepeated, String condition) Similar to DependentLoop(String, String, int, int, String, String) This method returns only one String where the different lines are separated by end-of-line char ("\n").
Vector<String>	**Dlookup**(String field, String type, String criteria) Dependent look up method.
String	**DlookupString**(String field, String type, String condition) Dependent look up String method.

SPECS INTERFACES METHODS

Vector<IDeviceInstance>	findMatchingInstances(String deviceTypeNames, String condition) This method is used to get all the instances of the specified device type(s) where the specified condition is true.
Vector<IDeviceInstance>	findMatchingInstances(String deviceTypeNames, String masterObject, String condition) This method is used to get all the instances of an specified device type with an specified master object, where the specified condition is true.
Vector<IDeviceInstance>	findMatchingInstances(String deviceTypeNames, String masterObject, String condition, List<Integer> positions) This method is used to get all the instances of an specified device type with an specified master object, where the specified condition is true.
String	GenericDepLoop(String instanceName, String targetDevice, String linkedField, int optionDuplicate, int first, int fStep, String fTextWithLink, String fTextNoLink) This function detects link between an object name in another object table (targetDevice) in a particular field (LinkedField). If a link is detected then it will replace in the string FTextWithLink the fields between # with the correspondent value in the target device and returns the string to the function. If several links are found, there are three different options: optionDuplicate=0 then it will concatenate the several strings; optionDuplicate=1 then it will keep the first replaced string; optionDuplicate=2 then it will keep the last replaced string; If no link is detected then, it will return the string FTextNoLink.
Vector<IDeviceType>	getAllDeviceTypes() Get all the device types available in the specs file.
IDeviceType	getDeviceType(String theDeviceTypeName) Get a device type from the specs file.
ISpecDocumentation	getProjectDocumentation() Get the project documentation data from the specs file.
String	getResourcesName() Get the name of the resources package used to build the specs file.
String	getResourcesVersion() Get the version of the resources package used to build the specs file.