■ Plugin API ■ IoX APIs ■ Configuration

Configuration

For a plugin or plugin to be understood by the IoX framework, it needs to have a set of **required** configuraiton files. You will create these files, package them up (zip format), and then Polyglot will upload them to IoX during installation.

Directory Structure

The directory structure is as follows:

`/nodedef` - A directory containing all the XML node definitions (.xml).

A Node Definition defines the status and commands available to a node. This is probably the most important file as it describes a node.

`/editor` - A directory containing all the XML editors files (.xml).

An Editor defines the parameters for a widget in the client, such as a combobox, a numeric field etc. It defines the set of values and the unit(s) of measure available. An editor may contain multiple `<range>` entries, each of which must have a unique UOM.

`/nls` - A directory containging all the NLS properties files (.txt).

NLS is a set of name/value pairs used to display values in national language (such as English). A single NLS file is used for each language. The naming convention is language_countryCode.txt (e.g. en_US.txt for USA English) N

`/linkdef` - A directory containing all the XML link definitions (.xml).

link definitions define the support for native links between nodes used in IoX scenes.

`/version.txt` - A file that contains the version of these files.

e.g.

```
/editor/edit.xml

/linkdef/ldef.xml

/nodedef/ndef.xml

/nls/EN_US.txt

/version.txt
```

Profile

A collection of files for plugin is called a `Profile`. Profiles correlate to slot numbers in IoX. So, once a zipped package for the plugin is created, it's uploaded to a specific profile/slot number is IoX and referneced that way. Here's an example of a list of profiles configured in IoX.

```
<profiles>
   <profile family="-1" id="1">
      <files dir="NLS">
         <file name="EN_US.TXT"/>
      </files>
      <files dir="EDITOR"/>
      <files dir="NODEDEF"/>
      <files dir="LINKDEF"/>
   </profile>
   <profile family="1" id="1">
      <files dir="NLS">
         <file name="EN_US.TXT"/>
      </files>
      <files dir="EDITOR">
         <file name="I_EDIT.XML"/>
      </files>
      <files dir="NODEDEF">
         <file name="I_NDEFS.XML"/>
      </files>
      <files dir="EMAP">
         <file name="I_EMAP.XML"/>
      </files>
      <files dir="LINKDEF">
         <file name="I_LDEFS.XML"/>
      </files>
   </profile>
   <profile family="3" id="1">
      <files dir="EDITOR">
         <file name="EDITORS.XML"/>
      </files>
      <files dir="NODEDEF">
         <file name="NDEFS.XML"/>
      </files>
      <files dir="NLS">
         <file name="EN_US.TXT"/>
      </files>
   </profile>
   <profile family="4" id="1">
      <files dir="NLS">
         <file name="EN_US.TXT"/>
      </files>
      <files dir="EDITOR">
         <file name="EDITORS.XML"/>
      </files>
      <files dir="NODEDEF"/>
      <files dir="EMAP"/>
      <files dir="LINKDEF">
         <file name="ZW_LDEFS.XML"/>
      </files>
   </profile>
   <profile family="6" id="1">
      <files dir="NLS">
         <file name="EN_US.XML"/>
      </files>
      <files dir="EDITOR">
         <file name="EDIT.XML"/>
      </files>
      <files dir="NODEDEF">
         <file name="NDEFS.XML"/>
      </files>
      <files dir="EMAP"/>
      <files dir="LINKDEF">
         <file name="LDEFS.XML"/>
      </files>
   </profile>
   <profile family="7" id="1">
```

```
      <files dir="EDITOR">
         <file name="EDITORS.XML"/>
      </files>
      <files dir="NLS">
         <file name="EN_US.TXT"/>
      </files>
      <files dir="NODEDEF">
         <file name="NDEFS.XML"/>
      </files>
   </profile>
   <profile family="8" id="1">
      <files dir="EDITOR">
         <file name="EDITORS.XML"/>
      </files>
      <files dir="NLS">
         <file name="EN_US.TXT"/>
      </files>
      <files dir="NODEDEF">
         <file name="NDEFS.XML"/>
      </files>
   </profile>
   <profile family="9" id="1">
      <files dir="EDITOR">
         <file name="EDITORS.XML"/>
      </files>
      <files dir="NLS">
         <file name="EN_US.TXT"/>
      </files>
      <files dir="NODEDEF">
         <file name="NDEFS.XML"/>
      </files>
   </profile>
   <profile family="10" id="7">
      <files dir="EDITOR">
         <file name="EDITORS1.XML"/>
      </files>
      <files dir="NLS">
         <file name="EN_US.TXT"/>
      </files>
      <files dir="NODEDEF">
         <file name="NDEFS.XML"/>
      </files>
   </profile>
   <profile family="10" id="24">
```

```
      <files dir="EDITOR">
         <file name="EDITORS1.XML"/>
      </files>
      <files dir="NLS">
         <file name="EN_US.TXT"/>
      </files>
      <files dir="NODEDEF">
         <file name="NDEFS.XML"/>
      </files>
   </profile>
</profiles>
```

Network Communications

> NOTE
>
> All the communications are seamlessly handled by Polyglot. So, as long as your node definitions follow the correct pattern, then everything else is taken care of by Polyglot

From IoX to the Plugin/Node Server

The REST API is used to communicate with a plugin when using a network connection. The IoX uses basic authentication with either http or https to communicate with the plugin. A custom base URL is also prepended to the REST command, allowing the plugin to customize the location of its REST support.

For example, if a base URL of /nodeservers/joe is configured, then the following URL would be sent to the plugin to query a node:

```
/nodeservers/joe/nodes/<nodeAddress>/query
```

Having a base URL also allows a device to support multiple plugins, each with its own unique base URL.

From Plugin/Node Server to IoX

The plugin must use basic authentication with either http or https to communicate with the IoX. It must also know the profile number the plugin has been assigned on the IoX because most REST API calls require this number in the URL. The IoX uses the profile number to ensure only the nodes owned by the profile can be modified, and to choose the IoX user number the plugin should be using.

For example, if the plugin has been assigned profile number 5, then something like the following URL would be used to update device status in IoX:

```
/rest/ns/5/nodes/n005_dimmer_2/report/status/ST/25.2/percent
```

Responses

When a Plugin receives a REST command, one of the following responses must be sent out immediately, before processing the request. The IoX will send a similar response after processing a request (see IoX REST Return Codes)

```
200 - HTTP_OK
    Valid request received, will run it
401 - HTTP_UNAUTHORIZED
    User authentication failed (the userid and/or password is missing or
incorrect)
404 - HTTP_NOT_FOUND
    Unrecognized request received and ignored.
503 - HTTP_SERVICE_UNAVAILABLE
    Valid request received but ignored because system too busy to run it
```