Build Projects

Build-Projects contain so called builds which help to automate the process of generating code and/or building the application. They allow to specify a series of script files which are to be executed and provide arguments which can be configured by the user.

- XML Definition
- Sample
- Editor
- Default variables
- Command Line Interface

XML Definition

Builds are defined in an XML Schema file which is located at “cgi_studio_connector/schema/BuildProject.xsd”. This file can be used to validate xml files and with some editors (e.g. Visual-Studio) and provide code-completion when writing the XML file.

BuildProject-Element

Root Element. Contains list of Root-Paths, global Variables and Builds.

Attributes:

Argument	Required	Description
version	Yes	Version of the Build-Project file.

Children:

Argument	Occurrence	Description
RootPaths	1	List of Root-Paths
Variables	1	List of global Variables. These Variables can be used in any Build.
Builds	1	List of Builds

RootPaths-Element

Defines a list of Root-Paths.

Attributes:
None

Children:

Argument	Occurrence	Description
RootPath	0 - *	Root-Path

Variables-Element

Defines a list of Variables.

Attributes:
None

Children:

Argument	Occurrence	Description
Variable	0 - *	Variable

Builds-Element

Defines a list of Builds.

Attributes:
None

Children:

Argument	Occurrence	Description
Build	0 - *	Build

RootPath-Element

Defines a Root-Path. Root-Paths can be referenced by variables to make them relative to a common path.

Attributes:

Argument	Required	Description
name	Yes	Name of the Root-Path.
description	No	Description of the argument.

Children:
None

Variable-Element

A variable for the build which will be applied for all executed scripts.

Attributes:

Argument	Required	Description

name	Yes	Name of the variable.
type	Yes	Type of the variable. Valid values are ‘Value’, ‘Option’, ‘Path’ and ‘File’
root_path_key	No	If set, paths (or files) will be relative to the given Root-Path.
value	Yes	Value of the variable. If the type is ‘Option’, valid values can be specified as a list of values separated by semicolons. If left blank, any value is considered valid.
description	Yes	Description of the variable.

Children:
None

Build-Element

A build which can be executed.

Attributes:

Argument	Required	Description
name	Yes	Name of the build.
description	Yes	Description of the build.

Children:

Argument	Occurrence	Description
ScriptPath	1	Path to the script files which should be executed for this build.
OutputPath	1	Output-Path for this build.
Variable	0 - *	Variable for this build

ScriptPath-Element

Path to the script files which are executed when this build is started. The script files should start with a number to indicate their execution order.

Attributes:

Argument	Required	Description
value	Yes	Relative path.
root_path	No	If set, path will be relative to the given Root-Path.

Children:
None

OutputPath-Element

Main output path for this build. Even if the build doesn't produce any artifacts or generates output into other directories, at least the log file will be saved here.

Attributes:

Argument	Required	Description
value	Yes	Relative path.
root_path	No	If set, path will be relative to the given Root-Path.

Children:
None

Sample

Below a sample XML file can be seen

```
<?xml version="1.0"?>
<BuildProject xsi:noNamespaceSchemaLocation="../../tools/ConnectorApp/Build/Schema/BuildProject.xsd"
                    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                    version="1">
   <RootPaths>
      <RootPath name="CONNECTOR_ROOT" description="Path to CGI Connector executable." />
      <RootPath name="PROJECT_ROOT" description="Path to CGI Connector project file." />
      <RootPath name="CGI_ROOT" description="Path to CGI root folder" />
   </RootPaths>
   <Variables>
      <Variable name="INTERFACE_GENERATOR_OUTPUT_PATH" type="Path" value="src/Interface" root_path="PROJECT_ROOT" description="
Path for generated output files." />
      <Variable name="INTERFACE_NAMESPACE" type="Value" value="Application::Interface" description="Namespace for generated Interface."
/>
      <Variable name="INTERFACE_DATAMODEL_INCLUDE" type="Value" value="Model/!DATAMODEL_CLASSNAME!.h" description="Include for
Data-Model." />

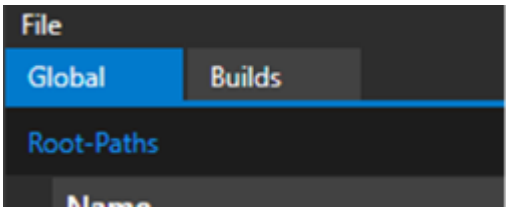
      <Variable name="BUILD_CGISTUDIO_PLATFORM" type="Value" value="!CONNECTOR_CGISTUDIO_PLATFORM!" description="CGI-Studio
Platform/Device." />
      <Variable name="BUILD_CGISTUDIO_PLATFORM_CONFIG" type="Value" value="!CONNECTOR_CGISTUDIO_PLATFORM_CONFIG!" description
="CGI-Studio Platform/Device configuration." />
   </Variables>
   <Builds>
      <Build name="Generate Code" description="Generates only code">
         <ScriptPath value="../BuildScripts/GenerateCode" root_path="CONNECTOR_ROOT" />
         <OutputPath value="" root_path="PROJECT_ROOT"/>
      </Build>
      <Build name="Generate and Build (Simulation)" description="Generates code and builds the application.">
         <ScriptPath value="../BuildScripts/BuildApplication" root_path="CONNECTOR_ROOT" />
         <OutputPath value="build" root_path="PROJECT_ROOT"/>
         <Variable name="BUILD_TARGET" type="Value" value="Application" description="Build target" />
         <Variable name="BUILD_CONFIGURATION" type="Option" value="Release;Debug" description="Build configuration/>
         <Variable name="BUILD_CMAKE_CUSTOM_FLAGS" type="Value" value="-DCOURIER_ADD_SCHOST_PROJECT=1 -
DAPPLICATION_IPC_ENABLED -DAPPLICATION_IPC_INTERFACE_ENABLED -DAPPLICATION_CONNECTOR_SIMULATION=1" description="Additional
CMake flags." />
      </Build>
   </Builds>
</BuildProject>
```

Editor

An editor for the Build-Projects definition can be found here:
“cgi_studio_connector/bin/BuildProjectEditor.exe”

If called with a file as first argument the editor will try to open the specified file.

Global



On this tab, global settings can be configured.

Root-Paths
Root-Paths allow the user to make path/file variables relative to a specified path. When a build is started these Root-paths must be set by the user beforehand.

They can be added and removed via the respective buttons on the right of the group.



Properties:

Property Name	Description
Name	Name of the Root-Path
Description	Description of the Root-Path.

Variables

Here, global variables can be set which apply to every build. However, each build can override the value of a variable simply by respecifying it directly in the build.

They can be added and removed via the respective buttons on the right of the group.



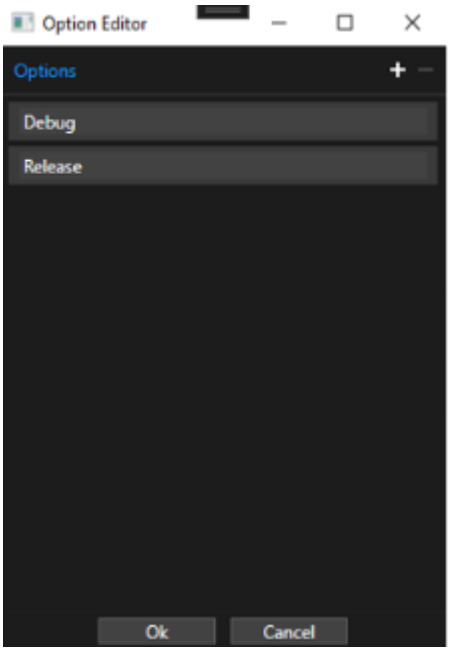
Propertise:

Property Name	Description
Name	Name of the Variable. This is the same name that is to be used in the scripts.
Description	Description of the Variable.
Type	Type of the Variable. The following options are available: - Value: Simple value type - Option: Value that must be set at run time. - Path: A path (folder) - File: Full file path
Root-Path	If the variable type is Path or File, this is used to reference a specified Root-Path. Doing so will treat the path as a relative path.
Value	Value of the variable. In case of 'Option' type values, the valid choices can be restricted by defining multiple values separated with a semicolon.

If the type is set to 'Option' the 'Option Editor' can be opened by clicking on the '...' button next to the Value field.



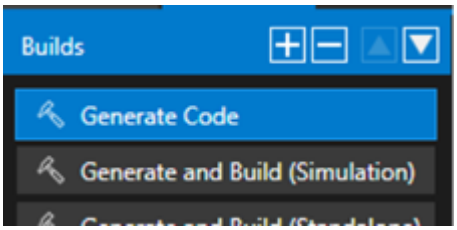
In the Option Editor window individual options can be easily added and removed using the 'plus' and 'minus' buttons on the right side.



Builds

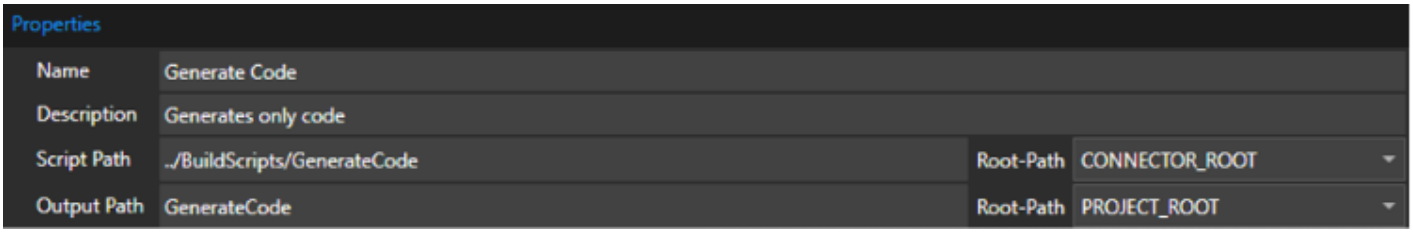
On this tab, individual builds can be configured.

On the left side, Native-Functions can be added/removed and reordered using the respective buttons.



Properties

Builds contain several general properties.



Property Name	Description
Name	Name of the build.
Description	Description of the build.
Script Path	Path to the script file which should be executed for this build. All scripts (.bat/.cmd) inside the specified folder are executed in alphabetical order so it is recommended to begin the filename with the number of the execution step.

Root-Path (Script Path)	Root-Path of the Script Path.
Output Path	Main output path for this build. Even if the build doesn't produce any artifacts or generates output in other directories, at least the log file will be saved here.
Root-Path (Output Path)	Root-Path of the Output Path.

Variables

Variables for this build can be set here.

They can be added and removed via the respective buttons on the right of the group.



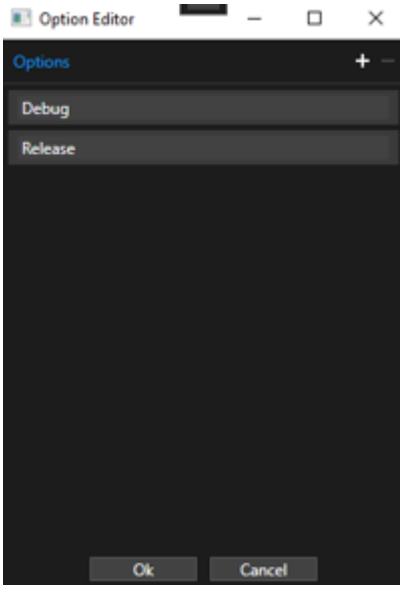
Properties:

Property Name	Description
Name	Name of the Variable. This is the same name that is to be used in the scripts.
Description	Description of the Variable.
Type	Type of the Variable. The following options are available: - Value: Simple value type - Option: Value that must be set at run time. - Path: A path (folder) - File: Full file path
Root-Path	If the variable type is Path or File, this is used to reference a specified Root-Path. Doing so will treat the path as a relative path.
Value	Value of the variable. In case of 'Option' type values, the valid choices can be restricted by defining multiple values separated with a semicolon.

If the type is set to 'Option' the 'Option Editor' can be opened by clicking on the '...' button next to the Value field.



In the Option Editor window individual options can be easily added and removed using the ‘plus’ and ‘minus’ buttons on the right side.



Default variables

Besides user configured variables the following default environment variables will be set.

Name	Description
BUILD_NAME	Name of the build
BUILD_DESCRIPTION	Description of the build
BUILD_SCRIPT_PATH	Path to build script files
BUILD_OUTPUT_PATH	Output path of the build

Command Line Interface

It is possible to start a build from the Command Line via the Build-CLI tool. For more information, please refer to the “Tools” chapter.