# Python-based Data Binding

The Player Connector Library is a simple library written in Python and c++ that provides functionality to start and load scene composer binary assets and interact with a predefined set of data binding items (Variant) through Python scripting.

# Player Connector Library Package

The main functionality of the library is contained in two files, **`CanderaAppConnector.py`** [1] and **`BindingSourceEnum.py`** [2]. These files contain three classes.

- `CanderaAppConnector`: This is the main class to be used to play assets and interact with them using data binding
- `BindingSourceEnum`: A Python enum generated by the build pipeline that contains the enum keys to be used when setting and getting data binding items. The keys underlay the nomenclature "SOURCE<*sourceIndex*>ITEM<*itemIndex*>". Source - and item index start at one and go to ten, as we currently have ten predefined binding sources and ten items per source.
- `Logger`: Logger class that my be used to generate logs with compliant format and logging levels

The sources and types listed above are contained in a file system Python package that can be installed with pip.

```
pip install path/to/python/library/root/folder
```

> It is suggested to install the library to a virtual environment and not to the global Python installation. For an exemplary usage refer to the Blood Pressure Sample documentation and source code.

[1]: */bin/ProfessionalEdition/Python/CanderaAppConnector/CanderaAppConnector.py*
[2]: */bin/ProfessionalEdition/Python/Generated/BindingSourceEnum.py*

## Workflow

The general workflow using the Player Connector Library package consists of:

1. Constructing the library with the Player Connector Library binary contained in the delivered package (PlayerConnectorLibrary)
2. Initializing the library with the asset binary
3. Optionally register data binding callbacks
4. Continuously calling step to invoke the render loop and internal message delivery
5. Interact with data bindings. Note that internally, changes to the data bindings are reflected on the view when step is called
6. Shutdown library and gracefully exit program

# Documentation

# Constructor

Creates and instance of this class. Expects and absolute or relative path to the Player Connector Library binary as parameter.

| | |
|---|---|
| Parameters | 1. `BinaryLibraryPath`: Absolute or relative path to the Player Connector Library binary |

# Init

Initializes the player with the provided asset path. Expects an absolute or relative path to the asset binary as parameter. Must be called before every other method. Returns a bool value whether the operation was successful.

| | |
|---|---|
| Parameters | 1. `AssetPath`: Absolute or relative path to the asset binary |
| Returns | The bool value |

# Step

Invokes the internal message processing and render loop execution. Needs to be called continuously. Databinding updates are reflected internally during the invocation of this method. Returns a bool value whether the operation was successful.

| | |
|---|---|
| Parameters | None |
| Returns | The bool value |

# SetIntValue

Sets an integer value to a predefined databinding source item.

| | |
|---|---|
| Parameters | 1. `BindingSourceItem`: Enum that specifies the binding source item. Enum is defined in *Generated/BindingSourceEnum.py*<br>2. `Value`: The integer value |
| Returns | The bool value |

# SetBoolValue

Sets a bool value to a predefined databinding source item.

| | |
|---|---|
| Parameters | 1. `BindingSourceItem`: Enum that specifies the binding source item. Enum is defined in *Generated/BindingSourceEnum.py*<br>2. `Value`: The bool value |
| Returns | The bool value |

# SetStringValue

Sets a string to a predefined databinding source item.

| | |
|---|---|
| Parameters | 1. `BindingSourceItem`: Enum that specifies the binding source item. Enum is defined in *Generated/BindingSourceEnum.py*<br>2. `Value`: The string |
| Returns | The bool value |

# SetFloatValue

Sets a float value to a predefined databinding source item.

| | |
|---|---|
| Parameters | 1. `BindingSourceItem`: Enum that specifies the binding source item. Enum is defined in *Generated/BindingSourceEnum.py*<br>2. `Value`: The float value |
| Returns | The bool value |

# GetIntValue

Gets an int value from a predefined databinding source item.

| | |
|---|---|
| Parameters | 1. `BindingSourceItem`: Enum that specifies the binding source item. Enum is defined in *Generated/BindingSourceEnum.py* |
| Returns | The integer value |

## GetBoolValue

Gets a bool value from a predefined databinding source item.

| | |
|---|---|
| Parameters | 1. `BindingSourceItem`: Enum that specifies the binding source item. Enum is defined in *Generated/BindingSourceEnum.py* |
| Returns | The bool value |

## GetStringValue

Gets a string from a predefined databinding source item.

| | |
|---|---|
| Parameters | 1. `BindingSourceItem`: Enum that specifies the binding source item. Enum is defined in *Generated/BindingSourceEnum.py* |
| Returns | The string |

## GetFloatValue

Gets a float value from a predefined databinding source item.

| | |
|---|---|
| Parameters | 1. `BindingSourceItem`: Enum that specifies the binding source item. Enum is defined in *Generated/BindingSourceEnum.py* |
| Returns | The float value |

## RegisterCallback

Registers a callback method. Multiple methods can be registered. Passed method must have two parameters. First param is  BindingSourceItem and second parameter is value.

| | |
|---|---|
| Parameters | 1. `CallbackFn`: Callable type with two parameters |

| | |
|---|---|
| Returns | The bool value |

# ShutDown

Shuts down the asset player and frees up allocated resources.

| | |
|---|---|
| Parameters | None |
| Returns | The bool value |

---

Revision #27
Created 2024-10-14 09:25:42 UTC by David.Aberl
Updated 2026-02-18 03:26:55 UTC by Kanai Tomoaki