

SmartBody Monitor

- [Deprecated](#)
- [Overview](#)
- [Users](#)
 - [Connecting to SmartBody](#)
 - [Resource Viewer](#)
 - [Data Viewer](#)
 - [Command Window](#)
 - [Face Viewer](#)
 - [Utils Tab](#)
 - [BML Creator](#)
- [Developers](#)
 - [Creating a Dialog](#)
- [Message API](#)
- [Known Issues](#)
- [FAQ](#)

Deprecated

NOTE - The SBMonitor is now deprecated, deferring it's functionality to the SBGui in Monitor Mode. Will have documentation for this soon.

Overview

The SmartBody Monitor (SBMonitor) can connect to any actively running [SmartBody](#) process in order to easily interact with and debug it.

Users

Connecting to SmartBody

1. Go to File -> Open (or hit Ctrl + C)
2. Wait 2 seconds for the dialog to appear
3. Select the desired [SmartBody](#) process to connect to and click the "Connect" button

Resource Viewer

Used for viewing all the resources that [SmartBody](#) is currently loaded, including .sk's, skm's, characters, services, events, face definitions, sequence and python files, paths, etc.

Data Viewer

Using a line graph to track channel values of a specific character's skeleton over time

Command Window

Facilitates the issuing of python, sequence, and [VHMsg](#) commands to [SmartBody](#)

Python Command format

```
<return value type> <python command>
int scene.getNumCharacters()
Possible return types: int, float, bool, string, int-array, float-array, string-array
```

Face Viewer

Allows the viewing and real-time manipulation of all AU's and visemes that are loaded for the selected character.

Utils Tab

Provides an easy to use interface for commonly used [SmartBody](#) functionality such as playing animations, setting postures, sending bml, gazing at targets, and text to speech commands.

BML Creator

Easily allows the creation of simple to complex BML commands through GUI buttons and sliders.

Developers

Creating a Dialog

1. Run https://svn.ict.usc.edu/svn_vh/trunk/lib/qt/bin/designer.exe
2. Create a new dialog and save it in [SmartBody/sbm-debugger/gui/designer](#)
3. Open `sbm-debugger\gui\SbmDebuggerGui.sln`. Create a new class that publicly derives from `QDialog` and have the header include `"ui_YourDialogName.h"`. If you compile, you will have compiler errors
4. Open a command prompt to [SmartBody/sbm-debugger/gui](#)
5. Enter the following commands**
 - a. `qmake -project`
 - b. `qmake`
 - c. `nmake`
6. Edit [SmartBody/sbm-debugger/gui/moc.bat](#) by adding the following line
 - a. `%MOC% YourClassNameThatYouJustCreated.h -o moc_YourDialogName.cpp`
7. Run [SmartBody/sbm-debugger/gui/moc.bat](#)
8. Copy `moc_YourDialogName.cpp` from the Debug folder into [SmartBody/sbm-debugger/gui](#)
9. Add `moc_YourDialogName.cpp` and `ui_YourDialogName.cpp` to the `vcproj` under the Generated folder

**You need to set the following environment variable for this to work

For using visual studios 2008

- QMAKESPEC win32-msvc2008

For 2010

- QMAKESPEC win32-msvc2010

Message API

Sends

- `sbmdebugger`

Receives

- `sbmdebugger`

Known Issues

FAQ

[FAQ](#)