

# Launcher

- [Overview](#)
- [Users](#)
  - [Using the Launcher](#)
  - [Using Profiles](#)
  - [Understanding Menu Items](#)
  - [Launching a program locally](#)
  - [Launching a program remotely](#)
  - [Launching and quitting multiple programs](#)
  - [Launching programs that do not conform to VHMSG messaging](#)
  - [VHMSG message protocol to work with the Launcher](#)
  - [Things to know](#)
- [Developers](#)
  - [Setting up the Launcher for a new project](#)
  - [Compiling the Launcher](#)
- [Message API](#)
- [Known Issues](#)
- [FAQ](#)

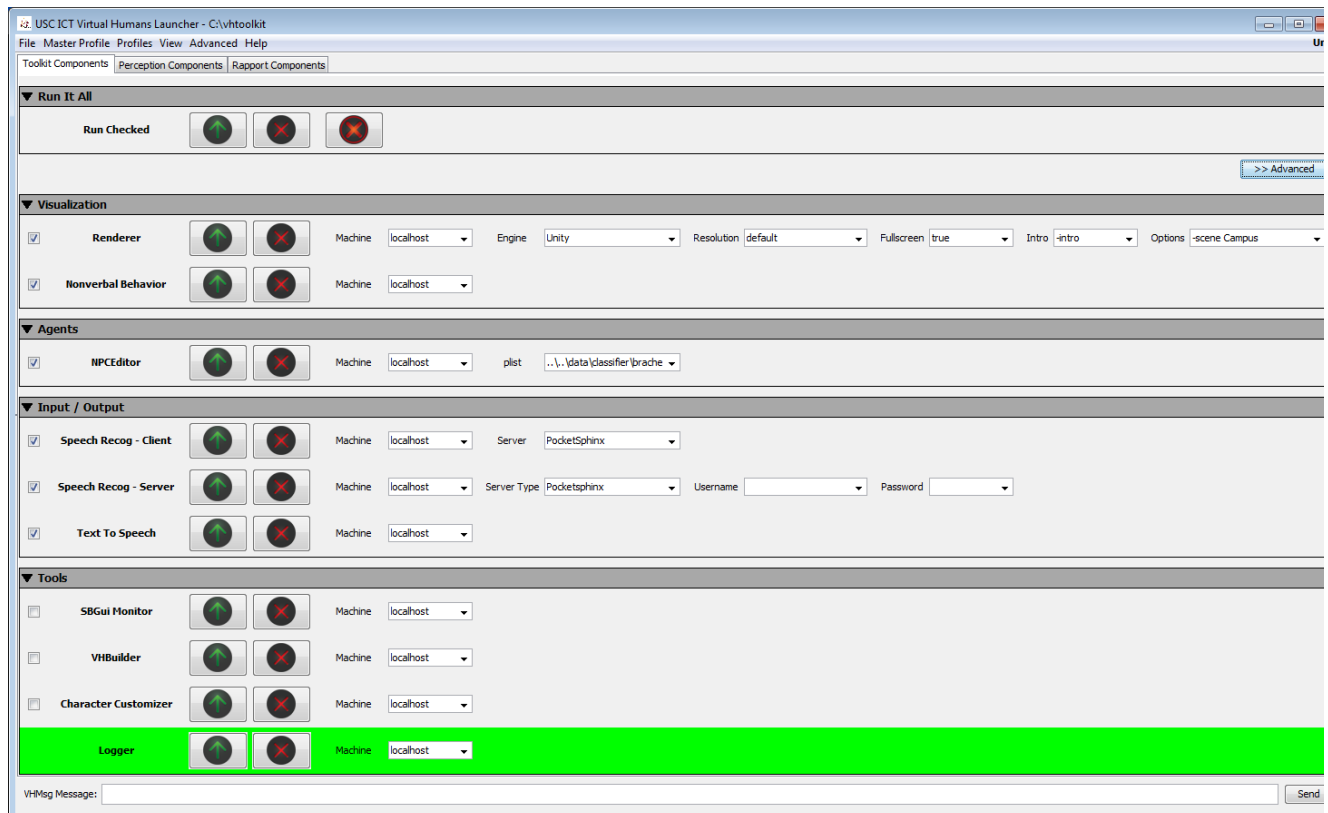
## Overview

The Launcher is a fully customizable tool for launching many kinds of distributed systems. The most notable features are:

- Allows tech leads to create Master Profiles (default system configurations) and users to create personalized profiles that reflect often used launch scenarios
- Shows visual overview of run-time module status, using VHMSG ping protocol; green lines means module is up and running; orange means module is unresponsive; grey means the module is offline.
- Launch work-in-progress SVN version of a component with a build version of other components or the full system, aiding developers in testing their work against a known working build before committing changes
- Launch / Kill components locally or remotely
- Run / Kill full system or individual components
- Send VHMSG messages

Quick facts:

- Project locations: /bin/launcher & /bin/launch-scripts (binary, used by default), /tools/launcher & /tools/launcher-scripts, /run-launcher.bat



## Users

Projects should have a run-launcher.bat file in the root of the project; this starts the Launcher with the correct Master Profile (the profile with all the default options).

Each component can also be launched manually from its associated row (called "launch tool") using the "Launch" button. Similarly, components can be killed individually with their specific "Kill" button. The "Run Checked" launch or kill button, will launch or kill all components with a check in the check-box of their individual launch tool. The "Kill All" button kills all the components (even the remote processes and the logger).

Notes:

- If there is no run-launcher.bat file in the root, look in \tools\launch-scripts. All batch files ultimately call run.bat in \tools\launcher\VirtualHumansLauncher.
- By default run.bat will load the example-master-profile.xml profile file in the tools\launcher\VirtualHumansLauncher directory. Other batch files can provide a parameter changing the master profile to be a project specific one.
- The Launcher will check whether a logger is online both on start-up and when a component is launched. If the logger is not online, the Launcher will launch the default Logger, that automatically starts logging all messages to disk. To disable this behavior, un-check the "Logger Always On" in the "Advanced" menu.

## Using the Launcher

Every project should start completely by clicking the Launch button in the Run It All section. Note however, that there could be dependencies between modules that make certain modules start before others. For instance, due to rendering issues, Unity may need to start after another component, such as AcquireSpeech, which can be added as a dependency in the Master Profile (explained in the section below).

The Run Checked "Launch" button will only launch selected modules and tools, as indicated by the check boxes next to each individual component. The Run Checked "Kill" button will only kill the same selected modules, while the "Kill All" button will send out a message to all components with the request to exit. To change which modules and tools will be launched / killed check or uncheck the check-box in each launch tool prior to clicking the Run Checked "Launch" or "Kill" button.

Click the "Advanced" button to toggle between the advanced and simple view. The simple view shows the Run It All group only. The advanced view shows the list of all the components and tools that can be launched. Use the advanced view to launch or kill components/tools individually and to make changes to a launch configuration. A component's row in the Launcher turns green when that component is launched and responds through a message that it is online. If the component fails to launch and/or respond, the row will turn orange. Reset the color status by right-clicking on the component name and selecting 'Set as not Launched'.

Some modules offer parameter options from a pull-down menu. [SmartBody](#), for instance, offers the ability to select which sequence file to load. Individuals may type in their own parameter values, which can then become part of the associated profile. See below for more information on profiles.

In addition to predefined profiles, individuals can create their own profiles. These can contain preferences, such as which components are checked or which parameters to use. Unsaved changes will be part of the [Default] profile. See below for more information on profiles. To change component information right-click on the component name and select the Edit option. New components can be added by similarly cloning an existing component. This usually only works with the [Master Config]. See below for more information. The View menu at the top of the Launcher offers several options to modify the interface, such as collapsing sections or parameters. The Advanced menu offers the option to turnoff periodic component status check messages (Enable vrAllCall). This might help when reviewing messages in the [Logger](#). Additionally, the Advanced menu offers the option of viewing the console, which shows the standard output of all components.

There are two important files associated with the launcher which are generated by the launcher:

- ApplicationConfiguration.xml - tells the launcher which files are considered as system profiles (explained below in "Launcher - Profiles" section). Additionally, go to this file to specify a different read me file for the launcher. By default the value is set to be "Virtual\_Humans\_Launcher\_Readme.txt"
- UserPreferences.xml - this file sets the profile which needs to be loaded on start up and is saved every time the launcher closes. It also defines other launcher options such as whether the logger is always on (whenever a launch tool is launched), etc.

## Using Profiles

There are 3 types of profiles (all are xml documents).

- Master Profile -  
This is the "main" file and typically is called master-config.xml. It contains all the information pertaining to the launcher and the launch tools. Additionally, this file is used to create the initial UI and, depending on which profile is selected, loads a specific configuration of tools.
- System Profiles -  
These profiles are typically listed in the file called ApplicationConfiguration.xml. These files were created by the developers of the system and they cannot be overwritten. If any change is detected in these profiles, the launcher will prompt "Save As..." and it will need to be saved as a different profile.
- User Profiles -  
These profiles contain customized values set in the arguments, but do not contain the actual launch tool's information. There is no limit to the number of User Profiles and they can be overwritten at any time.

There is also a "[Default]" profile, which is generated automatically based on the Master Profile. Any changes made to the "[Default]" profile are saved.

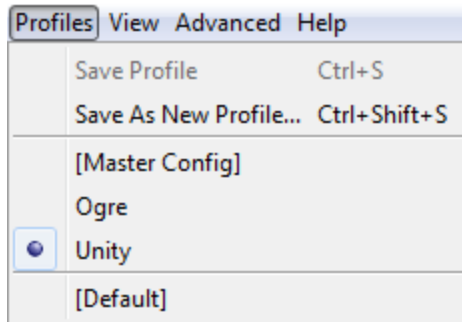
The Master Profile knows which folder to use for User Profiles. To select a User Profile listed in the "Profiles" menu, click on its name. The currently active profile has a circle next to its name in the menu, and it is also listed on the right side of the menu bar.

Example: To launch SmartBody with two different init files:

The Master Profile contains all of the information about how to launch SmartBody. Create User Profile 1 which specifies for the SmartBody launch tool to use the first init file. Create User Profile 2 which specifies for the SmartBody launch tool to use the second init file. Both newly created profiles will now appear under the bottom separator of the Profile Menu; click to launch.

## Understanding Menu Items

### Profile Menu



#### "Save Profile"

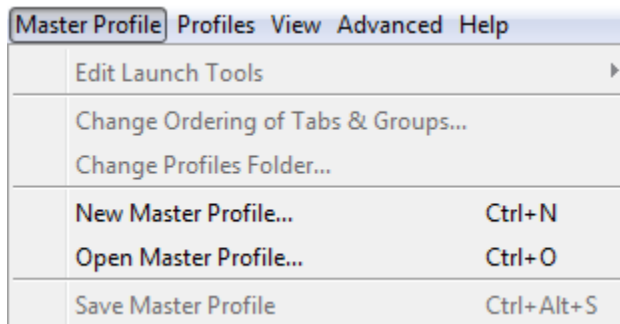
This option saves the current values of the arguments in all launch tools to the selected profile. This option is not available for the "[Default]" profile or any of the preloaded system profiles.

#### "Save As New Profile..."

Similar to the "Save Profile", this option saves the current values of the arguments in all launch tools, but instead of saving to an already created profile, it saves to a new one and prompts for a new name.

All of the loaded profiles appear below the separator. Click on the desired profile.

### Master Profile Menu



#### "Edit Launch Tools"

This menu contains many sub-menus for editing and/or launching tools. See the Creating/Editing Launch Tools section. This option is only available when the "[Default]" profile is selected.

#### "Change Ordering of Tabs & Groups..."

This brings up a window for reordering the tabs, groups, and launch tools from the GUI. This option is only available when the "[Default]" profile is selected.

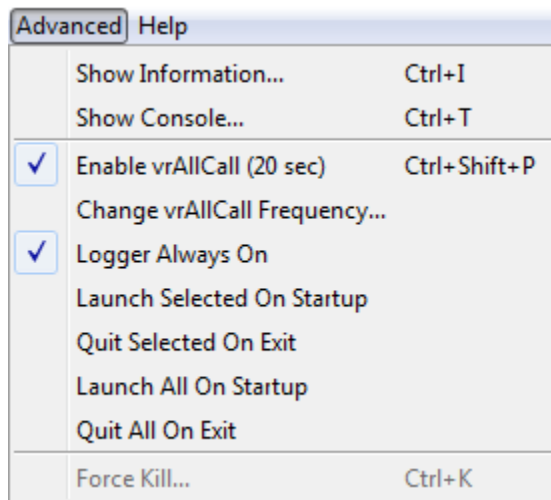
#### "Change Profiles Folder..."

This is for changing the location of the profiles folder for the currently loaded Master Profile to use. This option is only available when the "[Default]" profile is selected.

#### "Open Master Profile..."

This is for selecting the Master Profile. "Save Master Profile..." saves any changes to the Master Profile. This option is only available when the "[Default]" profile is selected.

### "Advanced" Menu



#### **"Show Information..."**

Brings up a window that displays miscellaneous information such as how many programs are running, which profiles are loaded, etc.

#### **"Show Console..."**

Brings up a window that shows the console output, as the standard console is suppressed by default.

#### **"Enable vrAllCall"**

This enables/disables the pinging ability of the launcher which sends out vrAllCall messages at the specified interval. If this is disabled, then the launcher won't be able to keep track of which programs are running.

#### **"Change vrAllCall Frequency..."**

Enables the ability to set the frequency of the vrAllCall messages mentioned above. Range is from 5 to 300 seconds.

#### **"Force Kill..."**

Brings up a window to force kill any programs that were started from the launcher on a local machine.

### **"File", "View", and "Help" Menus**

File->Quit - Quit the Launcher

View->Show - Arguments will show/hide the argument combo boxes and re-size the window.

View->Expand - All Groups will expand all of the groups.

View->Collapse - All Groups will collapse all of the groups.

View->Size - To Fit will re-size the window to fit its contents.

Help->Show - Readme will display a dialog that contains a copy of the readme (which is almost exactly the same as this wiki page). This can be configured to be a different file by setting a different value in the ApplicationConfiguration.xml. By default, the file is called - "Virtual\_Humans\_Launcher\_Readme.txt".

### **Creating/Editing Launch Tools**

Renderer

Name

Name:

VHMsg Name:

Organization

Tab:

Group:

Other Options

Selected Machine:

☒ Checked

Type:

Seconds:

Wait Component:

Mode: ☒ Build ☐ SVN

File

Parameters

Parameter Names

Engine

Resolution

Fullscreen

Intro

Options

Values

Reminder: Don't forget to save the profile if you want to keep your changes.

Name: The name of the program to display on the Launcher.

Elvin Name: The name that the program uses for elvin messages.

Tab: The name of the tab that contains the launch tool. To create a new tab, type in a new name and it will be created.

Group: The group that the launch tool is in. To create a new group, type in a new group name and it will be created.

Selected Machine: The default selected machine.

Type:

- normal: The normal launch tool, most programs use this one.
- agent: Use this when launching an agent.
- runall: Use this when creating a launch tool that runs the checked launch tools (see Launching Multiple Programs at Once).
- first\_wait: Use this when launching with a runall type to delay time before the next checked program launches. This is useful for programs such as the UT Server that takes time to start up. Multiple first\_wait programs can be used with each subsequent program launching after the time specified.
- logger = This is a special logging tool that is started automatically when the tool is launched.
- wait\_component = This makes the component wait on the specified component (whose value (elvin-name) is specified in the Wait Component). It's also possible to specify the time in 'seconds after the wait component is launched'.

Checked: If selected, the check-box will be defaulted to be selected.

Mode: Set either the "Build" mode or the "SVN" mode for the launch tool.

File: The file that should be launched to run the program. It will typically be an exe or a bat file. Use the "Browse..." button and it will automatically put it in the correct format.

Parameters: These are the parameters that will be shown on the launcher. Pre-set values are customizable. The "+", "-", and "Edit" buttons are self-explanatory.

Click the "Cancel" button to cancel without keeping any changes.

Click the "Delete" button (only when editing) to delete the launch tool.

Click the "Accept" button (only when editing) to commit all changes.

Click the "Create" button (only when creating) to create the launch tool.

Be sure to save a new Master Profile to keep all changes.

## Clicking on a Launch Tool

The screenshot shows a configuration window for a launch tool named 'Renderer'. The window has a light gray background and a black border. At the top, the 'Name' is 'Renderer' in bold. Below it, 'VHMsg Name' is 'renderer'. The 'Type' is 'wait\_component -- asr - [Component being waited on]'. The 'Mode' is 'build'. The 'File' is 'bin\launch-scripts\run-toolkit-renderer.bat'. The 'Machine' is 'localhost'. Below these fields is a section titled 'Parameters:' which contains a table with two columns: 'Name' and 'Current Value'. The table has four rows: 'Engine' with value 'Unity', 'Resolution' with value 'default', 'Fullscreen' with value 'true', and 'Intro' with value '-intro'. To the right of the table are three small buttons: a plus sign, a minus sign, and a list icon. At the bottom of the window is a 'Close' button.

Name	Current Value
Engine	Unity
Resolution	default
Fullscreen	true
Intro	-intro

Left click on a launch tool's name label to display a small dialogue-like window that contains information about the launch tool such as its elvin name (vhmsg name), its arguments, etc.

Right click to display the following menu items:

- "Edit <tool name>": Opens the Editing dialogue for the launch tool.
- "Clone <tool name>": Opens the Creating dialogue, but the new tool will be pre-populated with the data from this launch tool.
- "Switch to Build Mode": Makes this launch tool use the 'build' version of its program.
- "Switch to SVN Mode": Makes this launch tool use the 'svn' version of its program.
- "Set as not Launched": Sets this launch tool as not being launched.

## Launching a program locally

To launch a specific program locally, find the program on the Launcher and click on its pull-down list after "Machine:" to choose either "localhost" or the name of the local computer. Click the "Launch" or "Kill" button to either launch or close the program locally. If the program conforms to VHMsg messaging, its row on the Launcher will highlight in green when it is running. Note: If there is a VHMSG\_SERVER environment variable, it must be set to "localhost" or the name of the local computer.

## Launching a program remotely

This is done the same as launching a program locally, except select the name of the remote computer from the "Machine:" pull-down list. The remote computer must be running the [RemoteApplicationLauncher](#) on the same VHMSG\_SERVER and VHMSG\_SCOPE as the Launcher. See the readme file for the [RemoteApplicationLauncher](#) for more details. Make sure that the machine which runs the remote launcher has the environment variable VHMSG\_SERVER, which should be set to the machine which is running the Launcher. If not using remote launchers, this variable does not need to be set.

## Launching and quitting multiple programs

To launch multiple programs, check the boxes next to the programs to be launched and click the top "Run Checked" green "launch" button.

To quit multiple programs, check the boxes next to the programs to be quit and click the top "Run Checked" red "quit" button.

To quit all running programs, click the top "Run Checked" red "quit all" button. Note: This will quit all running programs whether or not their box is checked.

In addition, a Master Profile can be created to run other launch tools. This will launch any program with a checked check-box. If this is a first\_wait type of program, it will honor the delay (see above about Creating/Editing Launch Tools). Additionally, it will launch the programs on the machine that is selected in the machine argument of each launch tool.

## Launching programs that do not conform to VHMSG messaging

If a launched/running program does not support VHMsg messages, its row will highlight in orange not green. It will not respond to the "quit" or "quit all" buttons. This program needs to be closed manually instead of from the launcher. If the program is an exe, use the [ElvinAppWrapper](#) program which enable programs that do not support elvin messages to respond to the messages required by the Launcher. For more details, go to the [ElvinAppWrapper](#) readme file.

## VHMsg message protocol to work with the Launcher

The application needs to have an "elvin-name" (now called the "vhmsg-name") that it must use for sending and receiving messages. Note that the application's "elvin-name" needs to match the Master Profile's "elvin-name" tag for that application's launch tool. An application will only partially work with the Java Launcher, if it does not send and receive the proper messages. For more details about the messages go to the [VHMsg](#) page.

## Things to know

- If a launch tool is the normal gray color of the Launcher window, it is not running.
- If a launch tool is green, then it is running and sending the appropriate VHMSG messages.
- If a launch tool is orange, then it is running but there is possibly a problem. Either the program is not responding (i.e. it crashed) or it does not respond correctly to the vrAllCall message. The launcher can sometimes show a component as unresponsive in error, so it is advisable to check the status of the component manually if problems persist.
- Click on a group name to expand/collapse it.
- To send a VHMSG message, type the message into the text box at the bottom of the window and click the "Send" button. It has been tested with launching .bat files, .jar files, and .class files. Any properly formatted VHMSG should work.

For more information, contact the [vh-support@ict.usc.edu](mailto:vh-support@ict.usc.edu).

## Developers

### Setting up the Launcher for a new project

- Be consistent with the naming conventions below. Consistency between projects makes it easier for users and developers to get acquainted with a new project.
- Create a master profile, \tools\launch-scripts\profiles\master-config.xml, based on the provided example: tools\launcher\VirtualHumansLauncher\example-master-profile.xml.
- In \tools\launcher-scripts, create a batch file with the name 'run-<project\_name>-launcher.bat'. The content should be something like this:

```
@pushd ..\..\tools\launcher\VirtualHumansLauncher
call run.bat ..\..\launch-scripts\profiles\master-config.xml
@popd
```

- In the project root (usually the directory that contains the 'core', 'data', and 'lib' directories), create a 'run-launcher.bat' file. All projects should use this name, so everyone knows instinctively how to launch an unfamiliar project. The contents of the batch file should be something like:

```
pushd tools\launch-scripts
call run-<project_name>-launcher.bat
@popd
```

## Compiling the Launcher

Compile with either NetBeans or with ant.

Pre-Compiling:

1. Go to \lib\vhmsg\vhmsg-java\ and look for je4util.jar.  
If it does not exist, compile it by navigating to that folder in a command prompt window and typing 'ant' (no quotes). If that doesn't work, then install ant: <http://ant.apache.org>.
2. Go to \lib\java\ and check if it contains nblibraries.properties, win-process\winp-1.9.jar, win-process\winp-1.9-javadoc.jar, and swing-layout\swing-layout-1.0.3.jar.  
If it does not exist, then the project cannot be compiled.
3. Make sure that the environmental variable JAVA\_HOME is set to your JDK folder before continuing.  
Check the value of JAVA\_HOME by running the check-environmental-variables.bat file. If it is not defined, see the Pre-Running section for instructions on how to set it.

Compiling with NetBeans:

1. The folder that this file is in (\VirtualHumansLauncher), is a NetBeans project folder. Go to <http://www.netbeans.org> if NetBeans is not already installed.
2. Run NetBeans and go to File->Open Project and select the folder.
3. Click the button with the hammer picture on it (or Build->Build Main Project or F11).

Compiling with ant:

1. Open a command prompt window and navigate to the \VirtualHumansLauncher folder.
2. Type 'ant' (no quotes).

## Message API

Sends:

- [vrAllCall](#)
- [vrKillComponent](#)

Receives:

- [vrComponent](#)
- [vrProcEnd](#)

## Known Issues

- ...

## FAQ

See [Main FAQ](#) for frequently asked questions regarding the installer. Please use the [Google Groups](#) emailing list for unlisted questions.