Virtual User 1.0b4 Release Notes

This is a beta release of the Virtual User (VU) test automation system. This release package comes in the form of two 800K disks and four documents. The four documents are as follows:

1. **Virtual User General Reference: The first time user is suggested to start by reading this document. It describes the general architecture of the system and how to make effective use of it for test automation.**
2. **Virtual User Language Reference: This is a reference manual for the scripting language that has been designed to specify VU-driven tests.**
3. **Virtual User Release Notes (this document)**
4. **VU Recorder Reference Manual: This describes the sister application VU Recorder which can be used to augment the utility of VU. The Virtual User General Reference has a chapter on how to use VU Recorder in conjunction with VU.**

The two disks are as follows:

1. **Virtual User Program: This disk contains the Virtual User system which is comprised of the following main items. The Virtual User General Reference describes how to install VU on your machine using the contents of this disk.**
    - **Virtual User folder: VU, the MPW tool which runs the VU scripts, is located here. Other items in this folder are used to set up a VU environment in MPW.**
    - **Agent VU: This is the system Init which sits in the test machine's system folder.**
    - **Supplements: This folder contains other supplementary items like VU Recorder, special items needed to test MacApp applications, etc.**
2. **Virtual User Samples: This disk provides you with some example VU scripts and some example libraries to get you started. Each of the example scripts has a header which describes its purpose and the setup required to run it.**

VU Installation

To be able to use Virtual User you must start by reading the Virtual User General Reference document. This document describes the system, its installation procedure and other facts you must know to use VU effectively.

Things to note

- **MacApp Support**

  Virtual User has problems running against MacApp applications due to reasons described in the chapter "Running VU Against Applications built with MacApp" in the Virtual User General Reference document. We provide a Agent VU hook for MacApp applications to enable the application to become VU friendly. You will have to build your MacApp application with this hook to be able to test it with VU successfully. The source code for the hook and the accompanying document are available in the Supplements folder in the Virtual User Program

disk. A sample application built using this hook and a VU script which runs against it is available in the Virtual User Samples disk.

- **Popup Support**

  We have implemented support for popup menu selections and matching for those applications that use the CDEF (CDEF id = 63) which comes with System 7.0 or the one which the Macintosh Communication Toolbox uses. Please contact Apple Software Licensing regarding the use of the Communication Toolbox CDEF. It is not possible to retrofit applications that use the scheme outlined in Inside Mac vol. V.

- **Warnings for reserved word use**

  We are trying to safeguard against possible name conflicts with future versions of VU. VU will now warn against the use of a particular name (e.g. variable, task name) that may conflict with our future plans. This will not prevent you from using it now as there are no semantics tied to those words at present. If you want to know whether a particular script has any names that may conflict with the future without actually executing the script, you can use the compile only option (-c).

Known Problems

1. **Task definitions with task calls in the default values for their arguments do not work properly. For example,** `task test (arg1 := foo( )) begin…end;` **where foo( ) is a task call. The task call given as the default argument will always evaluate to undefined.**
2. **The grow trait for windows may be reported to be true when the window has no visible grow box. This occurs with windows which are technically document windows with grow boxes, but which choose not to draw the grow box. There is no known problem with the reverse case, which is to say VU will not indicate that a window lacks a grow box when it really has one.**
3. **VU will erroneously report that it has lost contact with its target if any matching operation is attempted while the target's mouse button is pressed in the menubar, a window's title bar, grow box, any control, or a menu selection is in progress. This problem arises when the target mouse is operated with the primitive mouse operation pressMouse and no corresponding releaseMouse command is given before matching is attempted or when some person holds the mouse button down while VU is operating the target. At present, there is no fix available to us for this problem.**
4. **There may be problems with situations where VU will be reading the items out of one dialog (with help from Agent VU) and a new dialog appears in front of the one VU is reading. VU continues to read dialog items, but it is getting them from the new front most dialog. The effect is to create a window descriptor whose content list is a mixture of both dialogs. An example where this might happen is with printer dialogs.**
5. **For reasons currently unknown the VU tool does not run on a machine running system 7.0. You can still test software running under system 7.0, as long as your host/controller machine which runs the VU tool is running system 6.0.x.**
6. **When you abort VU using cmd-. while a test is in progress, VU takes an extended period of time before it finishes its abort process. Part of this is due to the fact that VU has to abort any pending network transactions and do some other clean up.**
7. **Typed keys on target machines may be repeated when running on busy networks. This problem may be partially or completely elimintated by making sure that key repeat is set to off with the Keyboard control panel on each target.**
8. **The screen size of a target may be improperly reported if a system is transferred across machine types. For example, if a system running on a Mac SE had previously been running on a Mac II with a color monitor, the screen size of the color monitor will most likely be reported. If the system is transferred to a machine with a configurable monitor setup, this problem can be corrected by running the Monitors control panel. Otherwise, the workaround is to install**

systems specifically for the machine on which it will be running.

9. **Aborting a script during a typing operation on a target may cause the last typed key to remain in the key down position. It will remain that way until the key is typed again (manually or by VU).**
10. **Under System 7.0a9, the location of content items in control panel windows and in the Chooser will be skewed in relation to their real on-screen location. This will be fixed at least for control panel windows in upcoming developer-seeded versions of System 7.0.**
11. **If a variable (say, t) is assigned to a target descriptor, you will not be able apply the dot operator to access its zone trait (t.z). You can, however, unify the zone trait while matching to get the value directly as in `match [target z:?zone];`.**
12. **When using libraries, if your libraries use other libraries VU sometimes gets confused. Until this problem is fixed we suggest that in your main script file declare all the libraries that will be needed for your test. Please retain your library declarations in the library files as well so that when we fix this problem you can remove the redundant declarations from your script file without having to modify your libraries.**

Reporting Problems

To report problems, please use Outside Bug Reporter and/or link to MACDTS.

The following is most helpful when reporting problems:

1. **Version numbers for all components of Virtual User you are using. You can obtain the version of VU by using the -vers command line option.**
2. **The smallest VU script that shows the problem. In other words, try to eliminate as much of the script as you can. If it's one statement that causes the problem, send us just that one statement. Along with the script, describe the target state in terms of what application the script is running against and what state the app is in.**
3. **System configurations for both the target and the VU machine.**
4. **The network across which the test was being run.**

Contact

There is a Virtual User forum for you on AppleLink. Please post your questions here. The full path name to access it is:

Developer Services:Macintosh Development Tool Discussion:Virtual User