

*This session was presented at TAAC XI September 30, 2006. This was a two-hour hands-on session beginning with a brief discussion of troubleshooting guidelines, followed by Authorware programs that contained problem errors. As a group and within teams, the problems had to be identified, solutions found, and solved.*

---

### Troubleshooting Guidelines

#### Getting Feedback - Debug Tools

- Display Icon: Use the Display icon to display values of variables and functions.
- Debugger Structure: Create a Debug screen (display icon with embedded variables and functions) that can be activated at any time using a perpetual interaction icon. Use a hot key to activate the 'invisible' interaction.
- Control Panel -> Show Trace: to watch what is executing in the Trace Window.
  - o Trace function: Trace(\_\_\_\_\_) within a calculation icon. The result will display in the Trace window.
    - Copy expression from the Trace window and place in Notepad to look for a syntax problem
- Debug variables: History type variables to 'watch' the icons that have executed and in what order.
  - ExecutingIconID and Executing IconTitle: with the trace() function to see what icons are executing.
  - IconLogID(n) and IconLogTitle(n,m): will contain a list (Icon ID and Title) of the last given number of icons that have executed and the order in which they executed. Use in conjunction with the variable IconLog to see the number of icons to save (max 100)
- Use SystemMessageBox (function) like the Trace function but show values in a Window box.

#### Error Catching

- Start/Stop flags: Use these flags to start/ stop at a specific location in the program.
- Ctrl+B: Show Current icon (that is executing).
- Isolate icons: copy a portion of the icons to a new file, slowly add more icons until the error occurs again or the cause of the error becomes apparent. This is especially useful if your program seems to be corrupted.
- Create a log file: save values to an external text file along with explanations of when/what is happening in the program at the time the entry is made.

#### Authoring - Best Practices

- Comments: try to comment calculation icons when expressions are not obvious. Also use a calc with just comments as a README to explain program logic.
  - Title your icons – no UNTITLED icons. It helps you and others to maintain and debug your programs.
  - When trying to find the right variables/function to use, first start looking within the specific category if possible.
  - When combining variables and functions, break the function down into parts of the syntax to find any problems.
  - Backup: Keep multiple Backups
  - Don't get carried away with variables. Try to use the natural logic of the flowline, then as many of the built-in variables as possible if variables are needed. Also, try to use the natural flowline and placement of icons to control program logic.
  - Reuse calculations whenever possible. Put in a library or create your own command function.
  - When correcting a problem, don't get carried away with the corrections. Try one thing, see if it corrects it before trying something else.
- 
- 
- 
-

Common Areas where Errors Occur

- **Displays:** Display (changing) values of variables and "Update Displayed Variables" not set.
- When using **logical symbols** in expressions: := vs =, & (and), |(or), ~ (not)
- **Referencing an icon** (with the @) when the icon no longer exists.
  - o "Deleted Icon" appears in the calculation icon
  - o or when used in an expression and only evaluated at runtime: i.e. Eval("displayicon@"icon1\"). Icon removed and @ reference not modified.
- **Publishing:**
  - o Not including all xtras with final program. This usually occurs when there is an 'uncommon' xtra that is not always found with One-stop publishing.
  - o Manifest file:
    - Not including the line referencing the manifest file in the aam file.
    - Incorrect path to manifest file.
    - Not including all xtras and other externals (dll's, ucd's)
    - Length of aas files not matching actual files. Usually because aam is modified for manifest and saved in the incorrect location.
    - Recycle option omission could slow downloads
- **Referencing external files:**
  - o incorrect path to external file; incorrect external filename, case sensitivity on some servers.
  - o Net functions (ReadUrl, PostURL, etc.): incorrect syntax
- **Framework Icon and Navigation**
  - o Unused icons 'hidden' in the entry/exit pane.
  - o Looping from last to first page and vice versa.
  - o Pages not automatically navigating to the next page when they should.
  - o Navigating to the incorrect page of a framework.
    - Skipping pages or playing the same page repeatedly.
  - o Not returning to the calling icon/returning when it shouldn't.
  - o Not navigating to a page at all when it should.
- **Media**
  - o Video
    - Not starting/stopping.
    - Incorrect video playing
    - Video stopping too soon.
  - o Sound
    - Not starting/stopping.
    - Incorrect sound playing.
- **Interaction Buttons**
  - o Buttons displaying at the incorrect time.
  - o Buttons not disappearing/not appearing when they should.
- **Miscellaneous Interaction problems:**
  - o Target Area:
    - Item can be drug completely off screen and disappears
    - Incorrect drag items are matching target area
  - o Conditional responses are being matched too soon/too late.
  - o Time limit is matched too quickly.
  - o Event is never matched (i.e. when used with a Flash program).
  - o A keypress response is never matched
  - o Text Entry's match the incorrect values
  - o Scoring: Incorrect/correct, total questions, etc.
- **Jumping between other Authorware files.**
  - o Blank screen between files
  - o Variables changing (when they shouldn't) after returning to calling program.
- **LISTS:**
  - o Accessing entry incorrectly
  - o Property vs. Linear List
- **Miscellaneous**
  - o Erase Icon: Erasing the incorrect the objects, overusing the erase icon instead of natural erases incorporated into many of Authorware's icons.
  - o Using the incorrect system variable or function
  - o Switching syntax in a function (i.e. GetLine())
  - o Misspelling a system variable and so creating a custom variable.
  - o Referencing variables with " " (which makes the variable a text string).
  - o Not initializing variables or initializing in incorrect point in program (i.e. when passed from a calling program).