Tech Note 162
Script SDK Implementation Basics

Topic#: 000829
Last Updated: January 12, 2000

This *Tech Note* shows the basics of using the Wonderware® Script Toolkit to create your own custom Script functions. (The FactorySuite™ 2000 Toolkit is the only toolkit referenced in this *Tech Note*. Contact your local **Wonderware distributor** or Wonderware Technical Support for information on other Wonderware toolkits.)

Note This *Tech Note* assumes that you have some working knowledge with programming in C or C++. Also, we recommend that you use an ANSI C compiler when compiling the custom script function DLL.

Overview

The Wonderware Script SDK can be used to develop and embed custom functions or subroutines, which can be used in either an InTouch Script or expressions displayed on the InTouch application windows. The output from the Script Functions SDK is a **.wdf** file, which is used by the InTouch script engine to identify the exported function from the script DLL.

Note It's important that you understand the Windows NT DLL (Win32 DLL) and how to develop a DLL for Windows NT before attempting to using the Script Toolkit to create a script function.

Script SDK Basics

WindowMaker recognizes a script DLL by reading the associated .**wdf** file. The .wdf file is an encrypted file that specifies the calling sequence of the function, help file information, and paste link information. It's created from a Input Definition File (.**idf**), which is a simple formatted text file that's described in chapter 9 of the *Wonderware Extensibility Toolkit Users Guide*.

Once the .idf file is created, you need to convert it to a .wdf file by using the **Crypt** utility program (Crypt.exe, located in the Script SDK installation directory) by typing in the following command at the DOS prompt:

```
CRYPT <infile>.idf <outfile>.wdf /e
```

Where <*infile*> and <*outfile*> are the filenames of the .idf and .wdf files (example, CRYPT testfile.idf testfile.wdf /e).

You'll need to supply an optional Paste function that returns the syntax of the function and argument list for your users. The function declaration and implementation would look like this:

```
// Declaration
DWORD WINAPI PasteBuiltInFuncs ( LPSTR funcName, LPSTR result);

// Implementation
DWORD WINAPI PasteBuiltInFuncs ( LPSTR funcName, LPSTR result)
{
DWORD hilite=0;
Int I;

Lstrcpy(result, funcName);
If (lstrcmpi(funcName, "Actual_Function_Name") == 0)
{
```

```
lstrcat(result, "(help string description );");
hilite = MAKELONG(14,27);
}
return (hilite);
}
```

There are usually four files that compiled and linked together to make up a Win32 DLL:

ScriptMain.C – Provides the main entry/exit point functions

Scriptname.RC – Provides the resources for DLL

Scriptname.DEF – Defines the exports, Heapsize, Library, etc.

Scriptname.C – Code for the script functions (that is, the exported functions)

Once you've compiled and linked your script DLL, created the .idf file, and created the .wdf file, install your new script function library by copying the DLL and .wdf files to your InTouch installation directory.

Warning When your custom script function is executed, it's executed in the same process space as InTouch with your function having the control of the processor. Thus, if you have any errors in your function, it could freeze the system and cause WindowViewer to crash.

WindowMaker optionally calls two routines **WWDllInit()** and **WWDllFree()**, once when it tries to load the DLL and once before it frees the DLL. You don't need to supply these routines, but if they're supplied they can be useful if you need certain operations performed during the startup and/or shutdown of your InTouch application.

Using the IDEA SDK API with the Script SDK is a powerful and useful combination. This is especially useful when reading large amounts of data from an external source and generic functionality is needed in an InTouch application.

Conclusion

As shown in the previous sections, the Script SDK is easy to use and implement. The core of the SDK is to create a Windows NT DLL (Win32 DLL), along with an .idf file that specifies the calling sequence of the functions in the library. Then, using the Crypt utility (Crypt.exe) to create the .wdf file so that WindowMaker can recognize the existence of your script DLL by reading the .wdf file.

N. Patel

The *Tech Note* is published occasionally by Wonderware Technical Support. Editors: Sabrina Abercromby and Kelly Hauser; Publisher: Wonderware Corporation, 100 Technology Drive, Irvine CA 92618. There is also technical information on our software products at **Wonderware Technical Support**, our WonderFax fax-on-demand system at (949) 450-5050, and the Comprehensive Support Knowledge Base.

For technical support questions, send an e-mail to **support@wonderware.com**.

back to top