TYX Corporation

Productivity Enhancement Systems

Reference	TYX_0051_4
Revision	1.1
Document	MSDebug2.doc
Date	January 14, 2003



How to debug with MSVC++6.0

1. Studio Version:

This document is for Studio **1.17.0** and later. The pictures below have been made using version **1.21.2**.

The Subset used for the Atlas is IEEE716.89/Paws.

This document assumes that the user has some knowledge of Studio and MSVC++6.0.

2. How to debug the Wcem.dll from the MS compiler:

We can utilize Microsoft Visual C++ and debug our drivers.

The first step is to have an Atlas program and a device database. We are going to walk through an example in order to make this easier to understand. Here is the source file for a small Atlas and device database example:

```
------ Atlas1.atl ------
000000 BEGIN, ATLAS PROGRAM 'Simple Dynamic' $
E900000 OUTPUT, C'START TEST'
                                     $
   10 SETUP, AC SIGNAL,
     VOLTAGE 5 V,
     CNX HI
                                     $
999999 TERMINATE, ATLAS PROGRAM 'Simple Dynamic' $
------ end of Atlas1.atl -----
----- Ddb1.ddb ------
configuration Wcem_debugtest;
def, fnc, Dsp == 45; ** Display
begin dev FNG;
 cnx hi virtualpin;
 begin FNC = 10;
  control{
        voltage range 0 v to 140 v;
        }
  source ac signal;
 end;
end;
----- end of Ddb1.atl -----
The busconfi file should be as follows:
------ busconfi ------
   IEEE-488 Bus Configuration File -
;
"Channel"
            2
FNG BUS 2 MLA 11 MTA 11
------ end of busconfi------
```

Put those files in a Paws project called MSDebug2. In our case, we will put the project under C:\usr\paws\MSDebug.

You may place the **Busconfi** either in the local directory or you may place it in the station subdirectory in your Station.



• Build the project, as shown below, under the TYX Studio.

• Now that the device database is built, you can create a **Wcem** module as shown below. You can call the module **Wcem** and redefine the path to locate the module in the .paw subfolder rather than in .\Wcem.

Note: By default, when you add the name for the **Module Name**, the files will be put into Module name subfolder into the **.paw** folder, so you will need to delete the module name folder extension that gets added automatically in the **Module Location** edit box.

New	×
Files Modules	
Module Types:	
SWITCH DB 원: ITA DB	
CEM	
	Module Name:
	Wcem
	Module Location:
	C:\usr\paws\MSDebug2\
	OK Cancel

- You will then have access to the **Wcem Wizard** in order to create the source files that will allow us to build the **Wcem.dll**.
- Using the Wcem Wizard... under View, create the proper setup function associated to the Atlas Setup function. You will want to check the voltage parameter:

Wcem Settings	2
Device Interface General	
Resources	Parameters
Mapped Functions doFNG_10_Setup	Add
	<u>R</u> emove
ОК	Cancel <u>A</u> pply Help

• You may wish to add some features under **General** such as shown below. This will be responsible for an additional C++ file generated by the **Wcem Wizard** called **ctlr.c**.

Wcem Settings			×
Device Interface General			_,
Output	User Functions		
C C files	🔽 Interface Clear	Dolfc	
C++ files	Device Clear	DoDcl	
	🗖 Interrupt	Dolntr	
	🗖 Load	DoLoad	
	🗖 Unload	DoUnload	
	🗖 Open	DoOpen	
	🗖 Close	DoClose	
OK	Cancel	Apply Help	

- Once all the functions have been mapped properly, you should click on **OK**. The Wizard will generate a list of source files in the TYX studio. In this case, the list of files is the following: **Wrapper.cpp**, **key.h**, **error.cpp**, **ctlr.cpp** and **FNG.cpp**.
- The file that will include the driver code that you might want to debug will be in this case **FNG.cpp**. We will add a Display function in the **doFNG_10_Setup** function in **FNG.cpp** as shown below:

Note: If you build the Wcem.dll from the Paws Studio, make sure that you either delete it or overwrite it with **Wcem.dll** generated by the MS Studio, or it might cause some problems when debugging the **dll** from the MSVC++6.0 generated by the MS Studio.

3. The MSVC environment:

Now, we are ready to move on to the MS environment. This example uses the support of MSVC 6.0.

• From the MSVC++6.0 under <u>File</u>, chose <u>New...</u> and select **Projects**. Fill in the **Project name**, Project type and **Location** as shown below.

New			<u>? ×</u>
Files Projects Workspaces	Other Documents		
ATL COM AppWizard Cluster Resource Type Wizard Custom AppWizard Database Project DevStudio Add-in Wizard	🔊 Win32 Static Library	Project <u>n</u> ame: example Logation: C:\USR\PAWS\MSDEBUG2\	
 Extended Stored Proc Wizard ISAPI Extension Wizard Makefile MFC ActiveX ControlWizard MFC AppWizard (dll) MFC AppWizard (exe) New Database Wizard 		Create new workspace Add to current workspace Dependency of:	7
 Win32 Application Win32 Console Application Win32 Dynamic-Link Library 		<u>P</u> latforms: ☑Win32	-
		OK Ca	ncel

• After pressing **OK**, select **An** <u>empty **DLL** project</u>. As shown below.

Win32 Dynamic-Link Library - Step 1 of 1		
	 What kind of DLL would you like to create ? An empty DLL project A simple DLL project. A DLL that exports some symbols. 	
< <u>B</u> ack	Next > <u>F</u> inish Cano	el

- Press **Finish** and then **OK**.
- Now, select **<u>P</u>roject**, <u>A</u>dd to **Project** and <u>**Files**</u>.... This will open the window below. Select the TYX Cem C++ files one after another. Here, we will select **Wrapper.cpp**, **error.cpp**, **ctlr.cpp** and the **FNG.cpp** files.

Insert Files in	to Project	? ×
Look <u>i</u> n: 🔂	MSDebug2 🔽 🗲 🖻 📸 🎟 🕇	
Debug DeviceDB example ctlr.cpp error.cpp FNG.cpp	Nin32_Debug	
File <u>n</u> ame:	"Wrapper.cpp" "error.cpp" "FNG.cpp" "ctlr.cpp" OK	
Files of <u>t</u> ype:	C++ Files (.c), cpp), cxx), tlj, h), tlh), inl, rc)	el
In <u>s</u> ert into:	example	1.

- Press **OK**.
- We now need to add basically all the settings that you would have in the TYX Wcem settings. The first thing will be to include the additional path of all the header files that you wish to include in your project. In this case, one that is unavoidable is the **cem.h** file from TYX. The path is usually

Project Settings	? ×
Settings For: Win32 Debug	General Debug C/C++ Link Resourc
⊕ 🛱 example	Category: Preprocessor
	Preprocessor definitions:
	WIN32,_DEBUG,_WINDOWS,_MBCS,_USRDLL,EXAMP
	Undefined symbols:
	Additional include directories:
	C:\usr\tyx\include
	Ignore standard include paths
	Project Options:
	/nologo /MTd /W3 /Gm /GX /ZI /Od /I "C:\usr\tyx\include" /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /D "_MBCS" /D "_USRDLL" /D
	OK Cancel

C:\usr\tyx\include. This can be done going into Project->Settings->C/C++ in the Preprocessor Category as shown below:

• Note: For greater safety, make sure that the Use run-time <u>library</u> option under Code Generation is set to Debug Multithreaded as shown below.

Project Settings	? 🗙
Settings For: Win32 Debug ▼ ⊕ ∰ example	General Debug C/C++ Link Resourc Category: Code Generation <u>B</u> eset
	Processor: Use run-time library: Blend * Image: Debug Multithreaded Calling convention: Struct member alignment: cdecl * Image: Black struct st
	Project Options: /nologo /MTd /W3 /Gm /GX /ZI /Od /I "C:\usr\tyx\include" /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /D "_MBCS" /D "_USRDLL" /D
	OK Cancel

- Under the **General** Category in **Link**, we need to include all the libraries that will be addressed by our project. In this case, we need to add **cem.lib** and **user32.lib** at the end of the libraries already included as shown below. You may wish to delete any additional libraries that aren't used, but it will not affect your project if you leave them there.
- You also need to specify the location of the **Wcem.dll** output file. This is where you need to be careful about making sure that it will not conflict with the one generated with the TYX studio. In this case, we will just locate the **Wcem.dll** generated here in the same directory as the one used by the TYX studio, hence locally.
- Under the **Project Options**, we need to add /**def:**"C:\usr\tyx\include\WCEM.DEF" as shown below. If we fail to do this, the **dll** will build, but the Wrts will fail to make proper use of the Wcem.dll.

Project Settings	<u>? ×</u>
Project Settings Settings For: Win32 Debug	? × General Debug C/C++ Link Resourc(Image: Comparison of the section of th
	Project Uptions: /machine:1386 /def:''c:\usr\tyx\include\WCEM.DEF'' /out:'Wcem.dll''
	OK Cancel

- Under the **Input** Category in **Link**, we need add the additional paths for the additional libraries as show below.
- We also need to ignore the **libc.lib** library to avoid redefinition warnings.

Project Settings	? ×
Settings For: Win32 Debug	General Debug C/C++ Link Resourc Category: Input Object/library modules: le32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib cem.lib Ignore libraries: Ignore all default libraries libc.lib Force symbol references: Additional library path: C:\usr\tyx\lib Project Options: kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib oleaut32.lib uuid.lib
	OK Cancel

- The next step is to set up the environment so that the Wrts can be started from the MS studio. Under the **General** Category in **Debug**, you should look for the Wrts on your system for <u>Executable</u> for debug session. Typically, it's located under C:\usr\tyx\bin.
- Press **OK** on the **Project Settings** window.
- You are now ready to build the project and start it in debug mode. Make sure that the active configuration is the debug version, which is the default one (under **Build/Set Active configuration...**)



• Build the project by pressing **F7**.

🗱 example - Microsoft Visual C++ - [C:\usr\paws\MSDebug2\FNG.cpp]	<u> </u>
Eile Edit View Insert Project Build Tools Window Help	_ 8 ×
📔 😅 🖬 🗊 🐰 🖻 🖻 🗠 + 🗠 + 🖪 🎘 🚰 🅦 license 💽 🌇	
Globals) V (All global members) V 💊 doFNG_10_Setup V 🐼 V Search 🕮 🔏 ! 🗐	
<pre>#include "cem.h" #include "key.h" #include "key.h" #include "key.h" #include (limits.h) #include (float.h) //EEGIN{DFW}:FNG:10:0 int doFNG_10_Setup (double VOLT) //END{DFW} { Display("Entering doFNG_10_Setup \n"); return (int) 0; }</pre>	
<pre>Compiling Compiling Wrapper.cpp error.cpp FNG.cpp cl1.cpp Linking Creating library exampleWin32_Debug/Wcem.lib and object exampleWin32_Debug/Wcem.exp Wcem.dll - 0 error(s), 0 warning(s)</pre>	
Build (Debug) Find in Files 1) Find in Files 2) Results) SQL Debugging /	
Searches the online documentation Ln 9, Col 8 REC [COL] OVR	IREAD

- You can place breakpoints in the **CPP** files that you have under the MS Studio, such as at the Display function in the **FNG.cpp** file.
- Now you can run the project in debug mode via **F5** or via **<u>Build->Start Debug-> Go</u>**. You may see the following window if you missed the step regarding setting the **Executable for Debug Session**. The first time, you will see the following window

Executable For Debug Session	? X
Please specify the executable file for the debug session.	OK
Executable file name:	Cancel
•	

• That is the window that will determine which executable to launch that will load the **Wcem.dll**, which is the output of the MSVC project. Click on the arrow next to the edit box and select **Browse**. Select **Wrts.exe**. This exe is usually located under **c:\usr\tyx\bin**

Browse		? ×
Look in: 🔂	Bin 💌 🗢 🛍 🛗	
atstfc.exe		
paws.exe		
pawsinto.e	xe	
	r.exe	
So wrts eve	exe	
Txref.exe		
1		
File <u>n</u> ame:	wrts.exe Open	1
Files of <u>t</u> ype:	Executable Files (*.exe;*.com)	

- Click on **Open**, and then **OK**.
- Click **OK** on the message box that tells you that the Wrts does not contain debug information after checking the checkbox in order to not see this window the following time. This is the window that you will see if you didn't miss the step regarding setting the **Executable for Debug Session**.

Microsoft Developer Studio		
'C:\usr\tyx\Bin\wrts.exe' does not contain debugging information. Press OK to continue.		
Do not prompt in the future.		
OK Cancel		

• Once the Wrts started, you should select the MSDebug2.PAW project as shown below

Open		<u>? ×</u>
Look <u>i</u> n: 🔂	MSDebug2	 * 📰 -
	Win32 Debua	
MSDebug2	.PAW	
File <u>n</u> ame:	MSDebug2.PAW	<u>O</u> pen
Files of <u>type</u> :	PAWS Projects (*.PAW)	 Cancel

• Run the project from the Wrts and MSVC will stop at the breakpoint set in the **Setup** function as shown below. In this case, we placed a breakpoint in the **FNG.cpp** file.

🦇 example - Microsoft Visual C++ [break] - [FNG	.cpp]		
Eile Edit View Insert Project Debug Iools V	<u>/indow H</u> elp		_ 8 ×
📔 😂 🖬 🕼 🕺 🛍 🛍 🗠 – 🗠 – 🛛	🔁 🔉 😤 🉀 license	💽 🙀 📴 🕱 🖬 🕑 🔶 🤁	ው የ⊁ ቀን ቆ
(Globals) (All global members)	💌 💊 doFNG_10_Setup	💽 🖻 🔹 🎇 🖉	
<pre>#include "cem.h" #include "key.h" #include <float.h> #include <float.h> //EEGIN{DFW}:FNG:10:0 int doFNG_10_Setup (double VOLT) //END{DFW} { Display("Entering doFNG_10_S return (int) 0; }</float.h></float.h></pre>	stup ∖n");		1
			•
Context: doFNG_10_Setup(double)		X Name Value	
Name Value			
VOLT 5.00	0000000000	Watch1 X Watch2 X Watch3 X Watch4 /	,
Break at location breakpoint		Ln 9, Col 1 REC D	OL OVR READ

- As we can see, the value specified in the Atlas for the Voltage value is visible.
- <u>Note:</u> If you are having problem with the breakpoint, it will be because you will have built the Wcem.dll with the studio which has a release configuration and cannot be debugged. In order to overcome this problem, you need to do a rebuild all from the MSVC++6.0 compiler in order to overwritte the one generated by the Paws Studio.

4. How to debug the Atlas while you are debugging the Wcem.dll driver?

This is a simple procedure.

- 1.
- You need to build the Wcem.dll in debug mode. You need to start the Wrts from the MSVC++6.0 with F5. This will allow to debug the 2. Wcem.dll.

<mark>‰ example - Microsoft Visual C++ [run] - [FNG.cpp]</mark>	
Eile Edit View Insert Project Debug Iools Window Help	
📔 😂 🖬 🕼 🐰 🛍 🛍 🗅 🗸 🗠 🖌 🖪 🔁 🙀 license	💌 🐂 🛛 🗈 🖬 🗗 🔶 79 (P *0) 64
(Globals) (All global members) 💌 💊 doFNG_10_Setup	💽 💽 🗸 📓 🏝 🗶 🗐
<pre>#include "cem.h" #include "key.h" #include <float.h> //BEGIN{DFW}:FNG:10:0 int doFNG_10_Setup (double VOLT) //END{DFW} { Display("Entering doFNG_10_Setup `n"); return (int) 0; } </float.h></pre>	
Context:	Value
Name Value	Watch1 / Watch2 / Watch3 / Watch4 /
Ready	

3. Load the project that you want to run in debug mode.

Open			<u>? ×</u>
Look in: 🔁	MSDebug2	- 🔁 🖆 🎫	
COBJDIR Debug DeviceDB example_ Flow	Win32_Debug		
File <u>n</u> ame:	Type: Paws Document Size: 445 bytes MSDebug2.PAW	<u>O</u> per	1
Files of <u>type</u> :	PAWS Projects (*.PAW)	▼ Cance	

4. This will put the Wrts at the beginning of the TPS, ready to start. <u>Note:</u> Before you reach this point, the MSVC may have stopped at a breakpoint that you placed in the Wcem.dll code. This is not the case in this example.

Run T	ime System - Production (IEEE716.89/PAWS) [MSDEBUG2] w Run Control Window Help	
Run Run Reset Reset	Sistation Display Binary files "Atlas1", version 20021024 Date stamp: Wed Jan 15 17:45:31 2003 Built In LEX Information CEM 'C:\ust\paws\MSDebug2\WCEM.DLL', enhanced error reporting CEM Module User / Kernel Model 2 Version 20020701 (3.9.25)	Station Reset Test
	ULmt Meas LLmt	
For Help, J	press F1	11

Now, from the Paws Studio, go into Debug/Start Debug/Attach 5.

🚪 Paws Developer's Studio (ier	ee716.89/PAW5) MSDe	bug2.PA₩		×
He Edit View Project Build	Debug Options Windov Start Debug Stop Debugging	v Help Shift+F5	Go F5 Attach	<u></u>
MSDebug2 Project	Go / Run Break Execution	F5 Alt+F5	Remote	
PAWS PROJECT P	Toggle Repeat Toggle Breakpoint Remove All Repeats Remove All Breakpoints	F8 F9	BEGIN, ATLAS PROGRAM 'Simple Dynamic' \$ OUTPUT, C'START TEST' \$ STTUP AC SIGNAL	
Ddb1.ddb The second	Start at Cursor Run to Cursor	Shift+F10 Ctrl+F10	VOLTAGE 5 V, CNX HI TERMINEE ATLAS PROCEDAN (Simple Duramic) S	
Source Files	Step Over Step In Step Out	F10 F11 Shift+F11	IERMINAIE, AILAS PROGRAM "Simple Dynamic" Ş	
별 FNG.cpp - 열 Wrapper. 웹 Busconfi	Run ATLAS Step Run ATLAS Test Run ATLAS Block			
	Set Variable	•		¥ *//
Attaches to RTS running process				Ln 4 NUM //

6. This will lead to the following behavior in Paws Studio. In order for this to work you have to have the Wrts running. In this case, Wrts will have been launched from the MSVC++6.0.

Paws Developer's Studio (ieee716.89) File Edit View Project Build Debug Or	/PAWS) MSDebug2.PAW ptions Window Help	
	- A B ? D B B I I I I B B B B B B B	₹.
MSDebug2 Project	Bus Configuration: Local	
Paws Project Tools	; IEEE-488 Bus Configuration File -	
E BAWS PROJECT	"Che Atlas1.atl	
 ☐ ATLAS ☐ AtLAS ☐ DEVICEDB ☑ Dob1.ddb ☐ Weem ☐ Header Files ☐ Concernise ☐ Concer	FNG 000000 BEGIN, ATLAS PROGRAM 'Simple Dynamic' \$ \$ \$ \$ \$ \$ \$ \$ \$ 10 SETUP, AC SIGNAL, VOLTAGE 5 V, CNX HI \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$	
Variable Value	Module Statement / Line Procedure	
	Atlas1.atl 900000, line 2 SimpleDynamic	
Ready	Ln 3	

7. You may now debug both the Wcem.dll and the TPS at the same time: The Wcem.dll from the MSVC++6.0 and the Tps from the Paws Studio. When running the Tps, the execution of the Tps will be stopped at either breakpoints in the Tps or in the C++ code.