

# *SN Debugger for Windows*

## v2.2 of the Windows Debugger

*SN Systems Ltd*  
*Quayside*  
*40 Hotwell Road*  
*BRISTOL*  
*BS8 4UQ*

tel://+44 117 9299733

fax://+44 117 9299251

<http://www.snsys.com>

<mailto:john@snsys.com>

**Date: 2th September 1997**

(14\_0)

Welcome to v2.2 of the SN Windows debugger.

Check out the [Changes](#) section for differences between this version and v1.6 or v2.x. This document covers the Windows suite of tools and compliments the on-line documentation found in the psywin.hlp file.

### ***Quick differences between 2.13 and 2.12:-***

- Fix for missing structure tags fix within overlays for *Symmunged* symbol files
- Fix for crash when disassembling off the end of the memory map
- [Full differences](#)

***Make sure you don't miss out on updates;*** SN Systems host a number of mailing lists for sending out important information. Either check out the [web page](#) and look for 'support', or <mailto:john@snsys.com>. Also look at the debugger's home page ([web page](#) -> products -> debugger) for any fixes or updates to v2.0.

For information on the Plug-In API, see our [web page](#) or [mail us](#) for further advice.

## **Contents**

- [Installation Notes](#)
- [Articles](#)
  - [101 Reasons to use Symmunge](#)
  - [Integrating using the shell](#)
  - [The Debugger's Screens have gone Blank](#)
  - [Things you can Change](#)
- [Changes](#) (version and release changes and fixes)
- [Reporting Bugs](#)

## **Installation Notes**

## Systems DLLs

This release of the debugger uses three Microsoft system DLLs that have become part of Windows 95 operating system, but were not shipped with the original version of the OS in August 1995. The DLLs in question are COMCAT.DLL (implementing COM categories) REGISTER.DLL (for registering Plugin components) and an enhanced version of COMCTL32.DLL (implementing Windows common controls). Install will attempt to install COMCAT.DLL and REGISTER.DLL and register them if they are not already present.

COMCTL32.DLL will already be present on any system that has previously been installed with any of the following pieces of software:-

- Windows 95 ORP 2 (this is OEM Release Pack 2 - it is only available on some machines pre-installed with Windows 95 or on CD as part of the MSDN)

MS Internet Explorer 3.0 (this is available free of charge from Microsoft)

Office '97 and many of the products release by MS in 1996

COMCTL32.DLL is **not** re-distributable and so we are not allowed to ship it with this install. If COMCTL32.DLL is not present on your system then PSY\_WIN.EXE will not load and instead complain that there is an undefined dynalink in COMCTL32.DLL. To remedy this problem, you can obtain COMCTL32.DLL from Microsoft and also from SN Systems. Download and install the latest released version of Microsoft Internet Explorer. If you don't want Internet Explorer, you can then uninstall it.

## Project File Compatibility

*This note only applies if you have installed previous versions of the debugger on your machine, or want to share project files with other users of the debugger.*

v2.0 saves project files in a slightly different format from v1.x. This new format is not compatible with older versions of the debugger (1.x), i.e. v2.0 can read v2.0 and v1.x project files but can only save v2.0 files. v1.x can only read and save v1.x files. v1.x may crash on attempting to load a v2.0 file. If you share project files with other users of the debugger or think you may need to revert to a previous version of the debugger, then bear this in mind by saving copies of your project files before you let v2.0 convert them.

v2.0 will ignore any breakpoints you had set in a v1.x project.

## Articles

### 101 Reasons to use Symmunge

**Symmunge.exe** is the symbol file squashing tool that SN Systems have made available to developers for some time. Unfortunately, in earlier versions of the debugger, the loading time of symmanged files attracted a penalty caused by the fact that symmunge moves all structure tag definitions away from their variables to the start of the file. v2.01+ optimises loading time for symmanged files.

Now for the reasons you should use it:-

- Symmunge checks same-named structure definitions for inconsistencies. If there are any, symmunge warns you. Otherwise, symmunge throws the duplicate tag definitions away.
- v2.01 also throws away duplicate tag definitions on loading, but, if there are any duplicates, you are not warned. This extra check also takes time - each time your symbol file is loaded.

Normally, the compiler invents a name for anonymous structures. The compiler uses an enumeration based on the order in which the anonymous structures are encountered. This means that the same anonymous structure may get a different fake tag in different modules. It also means that fake tags are duplicated and position dependant in the symbol file. This all leads to symbol file bloat, increased loading time and heap wastage. Symmunge magically fixes all this in one fell pass.

**typedef struct { int member} MyTypedefTag;**

In case you were wondering when the compiler needs to generate an anonymous tag, the above definition line is a common example.

- Symmunge reconciles your typedefs to anonymous structures and removes duplicates. Normally, in the debugger you may see variables in watches as such:-

**struct\* pMyStruct = {...}**

Once symmunged, you will see:-

**MyTypedefTag\* pMyStruct = {...}**

In the expression browser 'type' view, you might normally see a pile of definitions for your structures and typedefs that are the same. Symmunge will remove these so you will only see one for each type.

*To use it, get symmunge from the BBS and add a line to your build batch file after the link.*

## Integrating using the shell

This article contains a number of tips that you can use to speed up the edit/build/debug cycle.

### Using file associations

You can use file extensions to associate files with actions. Under Windows 95, you can assign multiple actions to a single file type and then display them by right clicking on the file in an explorer window. In addition, the default action can be performed by double clicking on the file in an explorer window and the 'open' action performed by specifying a file as an argument to 'start' (see below).

In this example, the .bin extension is associated with downloading a binary image to the target at address 0xB0000000:-

1. Open an explorer window, goto view->options->File Types
2. Click New Type and enter 'bin' in the associated extension box.
3. Click New (action), give the action a name ('open' is needed to specify the action with the start command) and type in the following command line:-

```
"c:\psyq_win\bin\pqrn download %1 b0000000"
```

Followed by OK. You can then use the 'set default' button to set the action as the default behaviour for double-click. You can also optionally change the icon type for the file.

4. Now, double clicking on a .bin file will launch pqrn which will attempt to download it to the target.

CPE file and PSY files already have associations set up which you might care to modify.

### Launching the debugger with a new project

From v2.0 10\_8, the debugger allows only one instance of itself to be run. If you try to launch another instance, the first will either re-open its current project, or close its current project and load the one you specified on the command line.

The effect of this is that you can trigger the debugger to re-open a project from a third party program or from a make file.

### Using the 'start' command

The 'start' command is a Windows 95 shell program which looks for shell registered programs and files, and launches them. **psy\_win.exe**, **psyserve.exe** and **pqrn.exe** are all registered with the shell and so can be launched from the command line, from the run menu or from a make file.

Pqrn is perhaps the most useful since it can be used to download binary and cpe files. Otherwise, since PSY is registered with the debugger, and since the debugger re-opens or opens PSY files with the same instance, you can put 'start example.psy' at the end of a build to get the make program to kick the debugger into action.

### The Debugger Screens have gone Blank

This happens for one of several reasons. To understand which one is affecting you, read the notes below and also look for your target specific readme file.

- **The target is dead -**  
Whilst the debugger is running, something happened to the target which has stopped communications between it and the PC. Rebooting or resetting the target, either physically or through software usually brings it back to life.
- **The target hasn't booted an image -**  
Some targets require the debug stub to be initialised through a call to a library function. If this doesn't occur, or the target doesn't boot properly, then communications will be disabled. Ensure the target is booting correctly and you have a valid, debuggable image loaded.
- **The downloader is an older version -**  
Some targets have primitive downloaders built into ROM. The downloader can be replaced at reboot time and then held in memory either permanently by battery backed RAM, or until the target is rebooted, or until the power to the microcontroller is removed. All targets usually come configured so that the downloader is automatically upgraded when the 'reboot' command is issued, or when the windows programs are started. Nevertheless, if this isn't happening for any reason, it will cause this problem. Try checking the Ping ID of the downloader using the file server program or the diagnostic tool.
- **The program has disabled interrupts or not executed a pollhost() for a while -**

In this instance, the debugger's connection timed out. You can attempt a reconnect if it's likely the program re-enabled interrupts or entered a loop where pollhosts() are used.

- **You have manually disconnected using the disconnect command on the debug menu** - This might be useful if you want to ensure the debugger is not issuing polling ticks to the target. You can reconnect at a later date to continue debugging or stop the target.
- **The target is switched off** - Target hardware that uses a microcontroller probably requires the target to be switched on itself before a debug connection can be established.

In all cases, the debugger can remain running while it is disconnected from the target. In fact, that is why the screens go blank, rather than a modal dialog box appearing. With the connection disabled, you can safely save or load a new project, attempt a re-connect, or download memory into cartridge RAM.

## Things you can Change

The debugger stores configuration information in the registry. Some of this information is not modifiable using the debugger's UI but you can add or change registry settings to tailor the debugger to your needs. If a registry value is not present, the debugger uses a default value so you can change its behaviour by adding the name to the registry and specifying a different value.

All of these values are stored in the **HKEY\_CURRENT\_USER\Software\SN Systems\Psy-Q Debugger\Parameters** key

Value	Type	Default	Behaviour
IndicatorFontSize	DWORD	(Windows)	The size of the font used in indicator windows. (e.g. source file name)
IndicatorFont	STRING	(Windows)	The font used in an indicator window
IndicatorShowFileOnly	DWORD	FALSE	Show only the file name in the indicator in a source pane
IndicatorLowerCase	DWORD	FALSE	Convert the file name to lower case
ConfirmDownloadOnProjectOpen	DWORD	FALSE	Ask before downloading binary or cpe files during a project open
SymbolGrowth	DWORD	5,120	Triggers a re-scan to optimise memory allocation for a symbol file if the size has increased by this amount in bytes. If a scan is not done, file growth is used to estimate new

			requirements. A file bellow this size is always scanned.
ObjectsInFixedBlockDef	DWORD	30,000	Redundant as of v2.12
ObjectsInFixedBlockFun	DWORD	1,000	Redundant as of v2.12
ObjectsInFixedBlockBlock	DWORD	1,000	Redundant as of v2.12
ObjectsInFixedBlockFile	DWORD	100	Redundant as of v2.12
ObjectsInFixedBlockLine	DWORD	15,000	Redundant as of v2.12
AdvancedHeapMan	DWORD	TRUE	Redundant as of v2.12
AdvancedHeapPercent	DWORD	5%	Redundant as of v2.12
ExpressionPaneWidth	DWORD	512	Width of the horizontal scroll bar in characters when it is displayed in an expression pane
SourcePaneWidth	DWORD	1024	Width of the horizontal scroll bar in characters when it is displayed in a source pane
ShowLabelsInMemory	DWORD	TRUE	Show nearest label information in the status bar in a memory pane
CentreOnPC	DWORD	TRUE	Centre the pane on the PC in source and disassembly panes
ShowSLIInSourcePane	DWORD	TRUE	Show Source Level Information in the status line when in a source pane
OpenLastProject	DWORD	TRUE	Debugger opens the project that was being worked on last time the debugger was open.
RestoreWindowPosOnProjectOpen	DWORD	TRUE	Remember the position of the debugger's main window in a project file
AskBeforeClosing	DWORD	TRUE	Provide a warning when the debugger is being closed by pressing the 'x' button or by using the system menu.
			Show the indicator

IndicatorTop	DWORD	TRUE	window at the top of a pane (as opposed to the bottom)
DisassemblyPaneWidth	DWORD	256	Width of the horizontal scroll bar in characters when it is displayed in an disassembly pane
RememberFlashingCaretStatus	DWORD	TRUE	Remember if you switched the caret flashing status off when you exited the debugger, and stop it flashing the next time you start it
SaveCWDOnDebuggerClose	DWORD	TRUE	Save the CWD (current working directory) in the CWD value when closing the debugger
CWD	String	N/A	To do with the Save and Restore CWD parameters
RestoreCWDOnDebuggerOpen	DWORD	TRUE	Set the CWD to the value of 'CWD' when the debugger is opened
StopTargetBeforeProjectOpen	DWORD	TRUE	Attempts to stop the first unit of the target before opening (or reopening) a project file
ShowRunningIcon	DWORD	TRUE	Show an animated icon when the target is running
RestoreBreakpointsOnProjectOpen	DWORD	TRUE	Restore breakpoints that are saved in the project when the project is opened
SourcePaneGotoPCOnStopped	DWORD	TRUE	Goto the PC when the Esc key is pressed
SourceDelIsCtrlShiftJ	DWORD	FALSE	Mirror the DEL key with the operation of Ctrl+Shift+J - this is the 'back' button that deletes the current source file from the forward list.

Also, the File Server saves configuration information in the  
**HKEY\_CURRENT\_USER\Software\SN Systems\Psy-Q FileServer** key

Value	Type	Default	Behaviour
UseLastOpenDir	DWORD	TRUE	Restore the CWD to the last directory opened in the File Server

## Changes

The following lists includes bug-fixes and general user-interface improvements.

- [Changes from 1.6 to 2.0, 10\\_0](#)
- [Changes to 10\\_1](#)
- [Changes to 10\\_2](#)
- [Changes to 10\\_3](#)
- [Changes to 10\\_4](#)
- [Changes to 10\\_5](#)
- [Changes to 10\\_6](#)
- [Changes to 10\\_7](#)
- [Changes to 10\\_8](#)
- [Changes to 10\\_9](#)
- [Changes to 10\\_10](#)
- [Changes to 12\\_0](#)
- [Changes to 13\\_0](#)
- [Changes to 13\\_1](#)
- [Changes to 13\\_2](#)

### Changes from 1.x to 2.0 10\_0

- 'Stepping Mode'. Instead of defaulting to source level step from the source pane only, there is now a stepping mode which is indicated on the far right of the status bar. This can be either 'SRC' or 'DIS'. The debugger will change the mode to either SRC or DIS when either a Source pane, or a Disassembly pane become active. At any other time, the user can toggle the mode by pressing Ctrl+M. If the mode is SRC and a Source Level step cannot be performed, a Disassembly Level step will be performed instead.
- The contents of the expression clipboard is shown on the status line.
- When structures are opened, they are now displayed in the correct order.
- The 'z order' of views are now restored when a project is re-opened. The 'active' view and it's minimised/maximised state is also preserved.
- FP Status register appears in the register window.
- C bitfields are correctly displayed.
- Project file history is stored and displayed in the Project menu.
- Enumerations - both opening structures with them, and displaying the enum value.
- 'View Hot Key' switching - Up to 9 views can be assigned a hot key from '1' to '9'. To assign a

view to a hot key, press Ctrl+Shift+no. The number of the hot key assignment will appear in the title bar of the view. To then switch to a view using a hot key, press Ctrl+no. If the destination view is minimised, it will be restored or maximised, depending on the state of the current view. Hot key assignments are preserved in projects and in saved views.

- 'Modes for individual sub-expressions'. In v1.x, each expression pane (either local or watch) has a hexadecimal mode. In v2.0, you can set the mode of an individual sub-expression. The hot key "Ctrl+T" is used to toggle the mode of a sub-expression between decimal and hexadecimal. The hot key "Ctrl+Shift+T" is used to toggle the mode of the pane. All new expressions that are created or expanded inherit the mode of the pane. Changing the mode of the pane sets all sub-expressions to that mode, regardless of their current settings. However, root expressions in a watch window save their individual mode settings when the pane is saved or a pane in the same view is deleted. These operations are also available from the context menus of the watch and local panes.
- The INS key is now overridden by pane type. If a watch pane is active, the INS key invokes 'Add Watch'. Otherwise, INS invokes 'New View'.
- Add all locals to watch - This is accessed from the watch's context menu and the Ctrl+L accelerator.
- Step out of - This is accessed via toolbar button, Debug menu and the F12 accelerator key. The action unpicks the stack one level, calculates the PC at this level, sets a temporary breakpoint at this address, sets the stack level to 0 and then runs the target.
- Hotkey for re-open - Ctrl+R.
- Disable/Enable continuous update - Several improvements. First, the value 'polling rate' determines how often the debugger checks the exception state of the target. On some targets (e.g. Phoenix, Nintendo), checking the exception state has zero hit on the target. However, I don't recommend setting this value below 250 ms. Another value is used to specify the continual update rate. It is measured in polling ticks (e.g. 250 x 4 ticks == a continual update rate of 1 second). Continual update (although not polling) can be turned off altogether. The polling rate determines the granularity of discovering an exception when the target is running.
- Integers used in expressions are now shown in the mode (hexadecimal or decimal) of the expression itself. However, typecast expressions are always hexadecimal (because it is usually a typecast address) and subscript values are always displayed as decimal.
- The name of the project file is displayed in the main window's title bar.
- Memory Pane 'set maximum width'. The context menu for a Memory Pane has a 'Set Maximum Width' action which adjusts the width of the memory pane to fit the current size of the window. The action chooses a width which doesn't split any words below the current word size selected in the pane. The accelerator Ctrl+F10 also performs this action.
- When the 'x' button is pressed (or system->close), the debugger will ask for confirmation before closing down. This action can be disabled by adding a **DWORD** registry value "AskBeforeClosing" to the key "**HKEY\_CURRENT\_USER\Software\SN Systems\Psy-Q Debugger\Parameters**" and setting it to 0 (the default is 1).
- The hidden shortcut keys 'h' and 'x' have been removed from expression panes. Instead, Ctrl+T and Ctrl+Shift+T are used to toggle the display mode of an expression/pane. Array expand is activated using Ctrl+SPACE. There is no other GUI reminder of this accelerator. Array expand opens all the elements of an array instead of the first up to a maximum of 100 elements.
- The last few lines of all source modules can now be seen in a source pane.
- A horizontal scroll bar can be added/removed to/from source, disassembly and expression panes (watch and local) using the Ctrl+B accelerator. The state of this scroll bar is

remembered when the view/project is saved. For each pane type, the default appearance of the horizontal scroll bar is also set with Ctrl+B e.g. the last scroll bar toggle will determine if a new pane of the same type appears with the scroll bar. Watch and Local panes share the same default.

- The caret position in a memory and disassembly view is saved with the view. This also is saved and restored when another pane in the same view is deleted.
- When switching between address, disassembly and memory panes, if possible, the caret is positioned on the same address. For example, when switching from a source pane to a disassembly pane, the caret is positioned on the disassembly of the line that was under the caret in the source pane.
- Double Click toggles a breakpoint in disassembly and source panes.
- Toolbar buttons for stack back-trace. In addition, the Ctrl+PageUp and PageDown accelerators have been swapped round. This is so that they reflect the physical direction in which the stack level moves.

### Changes from 10\_0 to 10\_1

- 2 x GPF in psy\_win.exe to do with source panes fixed
- GPF in breakpoint manager when loading a project with a breakpoint pane fixed
- GPF in MipsCpu.dll occurring randomly fixed
- Install uses a different method for registering COM servers - hopefully more reliable

### Changes from 10\_1 to 10\_2

- 1 GPF in psy\_win.exe to do with the caret position in panes fixed

### Changes from 10\_2 to 10\_3

- File server menu slightly re-arranged. Uploader plug-in shipped and available from the uploader menu item in the filesaver
- History mutex problem fixed
- Debugger updated for big endian target support
- Psy\_co library updated to enable bios loading
- Bugs in the Psy-Q New Protocol SCSI driver fixed
- Download command added to pgrun.exe "download file.bin addressinhex"
- Physical breakpoints are now restored if removed by a target reset or download
- Check for valid frame pointer removed
- Mouse released when in the middle of a drag operation in a pane and the right mouse button is pressed

### Changes from 10\_3 to 10\_4

- **BPM:-**
- Puts BPs back if they have been removed by another component or another reset
- Responds to ambient colour and font change
- **Callstack:-**
- Remembers property settings
- Remembers state of expanded functions when the call stack level changes (but not when a target update occurs)
- Responds to font ambient property change

- **ExpMan:-**
- Edit watch now works
- Multiple selection now works
- Browser is much faster
- **Uploader:-**
- History is now listed in correct order
- **Setup:-**
- Allows full range of port IO settings for the SCSI v1.3 board. So does the VxD
- **Debugger:-**
- Stepping array element up and down bug fix
- Debugger adds multiple expressions in a watch window if selected by ExpMan
- **Miscellaneous:-**
- Symbol information removed from N64 debug stubs

### Changes from 10\_4 to 10\_5

- **DisMem:-**
  - New plug-in tool added to disassemble memory to a file.
- **BPM:-**
  - Breakpoints can be ordered by pressing colour headings in a BP pane
  - Nearest label is shown in BP pane
  - Expression is shown in BP pane
  - GPF when saving and loading conditional breakpoints in a project has been fixed
  - BP counters can be reset to 0 in the breakpoint dialog
  - Shift and double-click triggers a breakpoint edit in source and disassembly panes
  - support ambient font and colour changes
- **ExpMan:-**
  - Bug fixed which caused a GPF when using a symbol file with a multi dimensional array
  - ExpMan provides full information on array dimensions
- **Debugger:-**
  - Correctly updates arrays of structures when the array index is changed by '+' or '-'
  - Crash in BPM when project contains no BREAKPOINT2 stream
  - Double-click on a breakpoint causes source and disassembly panes to goto that breakpoint. Each disassembly and source pane has a new property 'Respond to breakpoint event' that can be toggled using the pane's context menu. When this property is off, the pane will not goto breakpoints that have been double clicked in a breakpoint pane. The property is saved with the view or project.
  - Anchor in a memory pane has been re-enabled. To set up an anchor, choose the anchor option from the memory panes context menu.
  - Goto in a memory pane didn't do anything if the destination address was already displayed in the pane
  - You can assign a hot key to plug-in panes. To do this, use the normal method to switch to a plug-in pane. With the pane active, press Alt+Shift+n where 'n' is a main keyboard number key from 0-9. If the assignment is successful, you will see a message box. Then, to switch another pane to be the assigned plug-in's type, press Alt+n. The hot key settings are stored with the debugger settings.
  - If you stop the caret flashing using Tools->Miscellaneous->, it can be started again by re-toggling this menu item. When the debugger closes, it will re-enable the previous

caret flashing rate. However, if the debugger is still running, switching to a different application will not re-start the caret flashing.

- An occasional GPF in goto in memory pane fixed.
- For big-endian 64-bit targets, downloading a cpe file didn't set the PC to the value set in the linker control file.
- You can position the cursor on spaces in a register window. The helps the scrolling of register windows that contain floating point numbers.
- The toolbar now disappears when it is disabled
- The updating problems on the toolbar when the window was resized have been fixed
- Shift and double-click now edits a breakpoint in source and disassembly panes (similar to Shift+F5)
- Some navigation keys have been added to source panes to help switching between windows of different types. Each source pane (on an individual basis) stores a history list of source files that have been viewed in the pane. The list is accessed by Ctrl+J (back), Ctrl+K (forward) and Ctrl+Shift+J (back with delete). Ctrl+Shift+J shows the previous source file in the list, but deletes the current source file from the forward list. The history list is stored with the pane in a view or project. If you insert a new module into your project (as opposed to adding one at the end), then the history list may be transposed when you next load the source pane.
- Ctrl+O in a source pane brings up the more source files dialog box
- In the more dialog, files are listed in alphabetical order, and double click in the list box triggers a switch to that file
- Occasional crash when page up was pressed in a source pane
- Using Ctrl+Shift+O to switch to a plug-in pane now brings up the plug-in manager dialog box immediately
- Debugger uses the previous size of a symbol file to tune the subsequent loading of the same project. See the heap management article for more information.
- Ctrl+Shift+I shows symbol heap usage.
- /minheap and /? command line options added
- **History:-**
  - A mutex problem causing the lock out of the file server and debugger if run at the same time
- **DOS Tools:-**
  - New version of RUN.EXE which correctly sets the PC for big-endian 64-bit targets.
- **Other Tools:-**
  - PsyLib2 fix when compiler -g option used
  - PsyDump.exe program shipped

### Changes from 10\_5 to 10\_6

- **Debugger:-**
  - Gpf when evaluating certain expressions (PSY\_WIN @ 0137:6000d551)
- **CallStack:-**
  - Copies parameters to the expression clipboard
  - Expands parameters to see derived values
  - Able to switch contexts when return type flag turned off
  - Remembers the state of expanded functions when the stack level changes
  - If there are two or more functions on the stack with the same name, double click always

- sets the level to the one you actually double clicked.
- Changes font when message sent by debugger.
- **Uploader:-**
  - Shows an hourglass if uploading / downloading of memory is time consuming.
  - Progress messages sent to message sink.
  - Bug fix - History list for binary file remedied

### Changes from 10\_6 to 10\_7

- **Debugger:-**
  - External loading of symbols now works from the unit menu. Symbols loaded this way are added to the current symbols (if any) and are not added to the project file
  - Reboot with a new project bug fixed. You can add files to a new project, choose reboot which will prompt for a file save name, then save and reopen the project
  - If the bios needs to be loaded on debugger startup, you no longer get the 'no recognised targets, repoll?' message
  - When the loading of a project fails (eg. wrong target type), the new project command on the file server is not greyed.
  - An anomaly when scrolling memory and disassembly panes using the scrollbar thumb has been fixed.
  - The hexadecimal mode in a local pane is now remembered when the pane is saved or a pane in a view deleted
  - A bug causing gpf when looking at low memory has been fixed. A number of different things could have triggered this bug including stepping through function prolog or expanding improperly initialised data structures
  - Viewing bitfields for big-endian 64 bit targets has been added
  - The reconnect to target command on the unit menu now works
  - A bug causing a gpf in MipsCpu.dll (0x602151b3) is now fixed. A number of different things could have triggered this bug
  - A gpf when parsing large expressions has been fixed
  - A gpf when opening a different project more than 8 times in a debugger session has been fixed
  - The evaluation of functions that return pointer types is now working correctly
  - The source pane 'more' dialog appeared twice, second time with nothing in it.
  - When the debugger is not connected to the target, a message is displayed in the built-in panes.
- **CallStack:-**
  - If expanding a context, collapse it and expand it again .. wouldn't show parameter.
  - More than one parameter, second parameter also a complex type . wouldn't show with a button to expand it.
  - Expanding an array type .. crash .. Due to way expression is being created
  - Problem with interpreting double click
  - Able to deselect parameter name and values
  - Show name and value property has been removed. These are always on by default, now
- **ExpMan:-**
  - Fix in Symbol Browser to show array details correctly. Now gives the number of dimensions
  - followed by the size of each diminution.

- Fixed problem in Symbol Browser when displaying very long symbol names.
- Cosmetic changes
- **Diagnostic Tool:-**
  - Added in this release, plus a program to run the check without loading the debugger
- **History:-**
  - Fixed a bug causing a gpf in HISTORY.DLL (0137:601f1462) when typing labels into the expression manager.

### Changes from 10\_7 to 10\_8

- **Debugger:-**
  - Stop the target running before open/reopen project. Can be disabled in registry - see things you can change.
  - Obscure GPF when repeatedly right clicking certain panes
  - CPE file properties -> Disabled choosing 'run after cpe downloaded' when 'download' not selected
  - Project Files -> changed accelerators to avoid a clash of keys
  - Project -> New - now asks for save confirmation on a changed project
  - Step out of could case the target to run if the return address appears to be 0xffffffff. If the PC is executing function prolog code, then the return address may be incorrect and the debugger will attempt to step to an invalid or incorrect address. The effect of this will be either a target run or a 'cannot position temporary breakpoint'. Both of these are unavoidable as it is impossible to tell automatically when the PC is in function prolog code.
  - No breakpoint -> Edit breakpoint -> Cancel - previously left a new absolute breakpoint in place. Now it is removed
  - Pressing SPACE in a memory pane does a go to on the last address (the one that causes the box to be drawn round it)
  - When opening a new project, the debugger lost and then regained activation, causing flicker and sometimes causing the debugger to disappear behind other windows.
  - Problems with not stopping at non-absolute breakpoints corrected
  - Now can't split panes less than 60 pixels wide/deep
  - Status line - better proportions
  - If moving the caret to the far left of a register pane window, the caret appears at the right.- fixed
  - Only one instance of the debugger can be run. If you re-run the debugger, the currently open debugger will attempt to re-open its current project file, or open the new one specified on the command line.
  - More dialog -> no files -> choose OK -> bang - fixed
  - Remember CWD in debugger and file server. These programs now default to remembering the local directory when they are closed, and restoring it when they are opened. To disable or modify this behaviour, see 'Things you can change'
  - Tool based plug-ins show the help string on the status bar when selected from the menu
  - Start pqrn and the file server from the tool menu
  - When entering a watch, you can now type cast an address to an array type and array dimensions are shown in the type for any array. For example, enter:- (int[3][4])0x80001020 allows you to browse 0x80001020 as though it were a 3x4 integer array.

- You can now typecast to and from int64. 'longlong' is also understood by the expression parser to mean int64
- Mapping of source pane lines to disassembly was not always correct. For example, pressing Shift+Tab didn't always position the PC on the expected line
- Goto in source pane and disassembly pane didn't move the caret if the destination address was already displayed on screen
- Update button and clear all breakpoint button on the toolbar. If you have upgraded a previous version, then these buttons won't be available unless you customise the toolbar.
- Allow the CentreOnPC global setting (made in the registry) be a property of individual panes for disassembly and source panes
- New help shipped and hooked
- Double click to expand, de-expand expressions in expression panes. Ctrl+double click expands arrays by opening all their elements (like Ctrl+Space). Shift+double click edits and expression (like enter)
- Remember the caret not flashing state when the debugger is loaded. If you turn the caret flashing off, it will stay off until you close the debugger when it will be restored. If you then open the debugger again, that caret flashing will be stopped again. Note, switching away from the debugger will not start the caret flashing again.
- Open debugger -> goto PC in disassembly and source windows
- Error message when loading symbol files with overlays
- Don't save 0 for fixed block sizes into the registry (or read 0) - causes the debugger to allocate all memory otherwise.
- Hook context pane messages into the status bar
- Pressing unit button when debugger offline - previously resulted in unexpected behaviour
- Re-enable editing of toolbar - alt-drag allows you to drag toolbar buttons around the toolbar or delete them. Double click the toolbar to customize it.
- Give focus back to window when app re-enabled - when the debugger regains activation, the focus is set back to the original pane.
- **Diagnostic:-**
  - Add 'help' button
  - Add Mnemonic shortcut keys to all buttons
  - Add 'check files' option
  - Give date and version numbers of system files & plugins
  - Ping the target when doing a hardware test
  - Output the Plugin registry details in a registry check
- **Callstack:-**
  - Parameters expanded to see derived types, stepping or running machine would collapse the tree control.
  - No longer displays "Call stack not available" when target is running, stack display is updated when target stops.
  - Support for help via F1 key.
- **File Server:-**
  - CWD is remembered when the file server is closed and restored when it is next opened. See Things you can Change for details how to disable this.
  - You can specify an fixed or the CWD directory for file serving. This is the directory which relative file names will be searched for when an PCopen or PCcreate call is

made. See the main menu for more details.

- **DisMem:-**
  - The wait cursor is set when the disassembly is being done
  - Forwards progress messages to the output pane
- **Uploader:-**
  - Help support via either the help button or the F1.
- **Breakpoint Manager:-**
  - Store Breakpoint relative to nearest assembler label when a sym file is loaded
  - Pass WM\_MENUSELECT to parent within pane
  - Fix Accelerator keys in Dialog (insert and delete to add & remove breakpoints)
  - Add Help and Mnemonics to dialogs
  - Always have a breakpoint selected in BP Dialog and Pane
  - Fix bug causing deleted breakpoints to be restored undeleted
- **Plugin Manager:-**
  - Add PsyQBuildComponent category
  - Implement Help
  - Add Mnemonics to buttons
- **ScsiDrv3.vxd:-**
  - Fixed the port address sanity check. This was previously failing for some valid IO addresses causing the system to appear to not find the SCSI card at that address.
- **PQRun:-**
  - Always returns success so that it won't terminate a make file if invoked as a build step. This includes if the specified operation fails.

### Changes from 10\_8 to 10\_9

- **Debugger:-**
  - GPF occasional occurring when the symbol file is unloaded
  - Windows menu accelerators now work when a main debugger pane is active (eg. Alt+V brings down the View menu)
  - MIPS targets: bit 27 of SR was erroneously being checked to enable display of odd FP registers. Should have been bit 26.
  - A problem setting breakpoints when the target is running has been fixed
  - If setting 'run after cpe downloaded' flag for the project files, the target was run before any breakpoints were set.
  - The window icon changes when the target is running. It is also animated when running, this can be turned off... see the 'things you can change' article.
  - Help includes a link to this readme file and SN Systems' Home Page
- **File Server:-**
  - Occasional crash when file server starts for the first time (DEX targets)
- **PQRun:-**
  - 3 commands added:- **tstart**, **tstop** and **halt**. These start, stop and halt (makesafe - disable interrupts) the target when issued.
  - GPF when downloading a CPE for a second time.
- **Breakpoint Manager:-**
  - Fix for restoring breakpoint relative to assembler labels

### Changes from 10\_9 to 10\_10

- Fix for PClseek on some targets
- Fixed opening a project while the target is disconnected replaces all BPs with NOPs. Bug introduced in 10\_9.
- Miscellaneous changes

### Changes from 10\_10 to 10\_11

- N/A

### Changes from 10\_11 to 12\_0 - v2.0 to v2.01 beta

- **Debugger:-**
  - Registry option to discard breakpoints on project open. The setting is RestoreBreakpointsOnProjectOpen and the default is '1'. For more info, see 'things you can change'
  - Horizontal arrow keys, accompanied by 'shift', move the horizontal scroll bar thumb left and right. If the pane doesn't have a horizontal bar, Ctrl+B will make one appear.
  - Indentation in watch and local panes can be switched on/off. To do this, right click and toggle the menu option. The depth of the expression will be indicated in decimal on the left.
  - The display of deeply nested arrays is now much less verbose. Previously, the display contained the full C name of the expression listed from the root of the expansion.
  - In expression panes, strings are shown for (unsigned) char\* and (unsigned) char[], up to a maximum of 0x1ff characters with the strlen being displayed at the end of the string. In decimal mode, the string is shown in raw format for 128 characters and without terminating at a null.
  - Ctrl+J will remember the caret position of the previous pane. Ctrl+J is like pressing the 'back' button in a browser and goes back to the previously displayed source file.
  - The Source pane doesn't snap to the PC when the target is stopped using Esc. Instead, Space must be pressed. The registry setting SourcePaneGotoPCOnStopped can be set to TRUE to override this behaviour.
  - The Space key no longer goes to the last 'goto' result in a memory pane. Previously, this meant that entering space in the ascii portion of the pane was awkward.
  - Ctrl+J and Ctrl+K now cycle when the end of the list is reached.
  - The Del key can be set to mirror the Ctrl+Shift+J key. The registry setting SourceDelIsCtrlShiftJ can be set to 1 to enable this.
  - Search in a source pane:- Initiate search using Ctrl+Shift+F - this dialog box has a 'find' history. Ctrl+F does a find next. Finding will move the caret to the line that contains the string.
  - Optimisation of watch lookup time and symmunge symbol file load time. It is now advantageous to symmunge your symbol file at the end of the link process. This will reduce loading time, heap usage and improve structure and typedef consistency within the debugger. See [101 Reasons](#) for more information.
  - Ctrl+D will download the files that have been flagged for downloading at project startup - but without reloading the symbol file (as Ctrl+R).
  - Various bug fixes
- **Call Stack:-**
  - now evaluates parameters in correct context, able to handle recursive functions and fixed problem of GPFs.

- **File Server:-**
  - Fixed copy to clipboard bug
- **Expression Manager:-**
  - The Symbol Browser is now separate component from the Expression Manager, and is available as a view pane type and as a separate plug-in tool. As a result, the Browser plug-in must be installed to allow symbol completion and selection from the Expression Manager.
  - The Symbol Browser will now display and highlight the first symbol which starts with a character typed from the keyboard.
  - Column widths in the symbol list of the Browser now automatically resize proportionally to the size of the Browser's window.
  - Bug fix in Browser to display a symbol's Type data correctly.
- **Breakpoint Manager:-**
  - Columns in the Breakpoint Manager Pane no longer resize automatically.
  - Fixed a bug which caused Disabled Breakpoints to trigger in certain circumstances
  - Breakpoint Manager Pane now remembers your Font Selection
  - Fixed a bug causing Breakpoint Manager Panes to appear active when inactive.
- **MipsCpu.DLL:-**
  - Showing odd fp registers on MIPS 4\*
  - BC1F stepping bug fixed

### Changes from 12\_0 to 13\_0 - v2.01 to v2.11 beta

- **Debugger:-**
  - Full automatic support for overlays from v2.10 of the debugger
  - PIO interrupt support has been added for the DTL-H2500. This allows the debugger to communicate with the H2500 without the need for a pollhost() call. When this feature is enabled, a PIO interrupt is generated on the PSX by the driver everytime a command is sent to it. This means that the H2500 is forced to service all requests as and when they are generated by the debugger and file server. The PIO interrupt is selectable from the *DTL-H2500 Communications* plug-in from the *Tools* menu. Simply toggle interrupts by clicking on the check box.  
This PIO facility also allows the debugger the ability of recovering communications should the PSX stop responding. As soon as the H2500 doesn't respond to a command the debugger will display a dialogue asking if you wish to attempt a recovery. Selecting 'Yes' will temporarily enable PIO interrupts and halt the target if it can. 'No' will ignore the fault this once and try again to reconnect again, and 'Cancel' will ignore the fault and not attempt a recovery even if the target fails to respond.  
**Note** that the PIO Interrupt is **not** an NMI, so will only function as described **if interrupts are also enabled on the PSX.**
  - Fixed stepping last line in a source file bug.
  - Fix crash in loading certain long symbol files.
  - Remove "this" being added to the Windows clipboard feature :)
- **File Server:-**
  - Added *DTL-H2500 Communications* plug-in to Tools menu.
- **Psy\_Co.DLL:-**
  - Fixed error on opening read only CPE files.
  - Removed unneeded extra wait when rebooting the H2500

- **Uploader plug-in:-**
  - Optimised heap requirement for downloading large binary files.
  - Remembers upload/download radio button state.
  - Remembers last file name and last download/upload addresses without having to use the history list.
- **Symbol Browser plug-in:-**
  - Fixed the plug-in grabbing window focus when function scope changes.
- **DTL-H2500 drivers:-**
  - Fixed some ASPI stability problems with the H2500.

### Changes from 13\_0 to 13\_1 - v2.11 to v2.12 beta

- **Debugger:-**
  - Memory estimation and allocation for symbol file loading has been greatly altered. The debugger now makes use of *scanning* and *scaling* before loading symbol files.
    - *Scanning* - This parses the symbol file so that the memory requirements for the file can be established before loading its contents into memory. However, Defs are estimated to reduce memory usage because of duplicates which are normally discarded after loading.
    - *Scaling* - If the symbol file has been previously loaded and saved as part of a project, then the debugger can estimate memory requirements based on the amount the file's size has changed. This shortens load time by removing the need to scan the file.

Which method is used is dependant on whether a symbol file has been previously saved, and if it has then by the amount its size has changed. If the file is new to the project, then it will be scanned when the project is opened. If this is not a new file, then the size difference from its previous size is compared to the value **SymbolGrowth** stored in the Registry (see [Things you can Change](#)). If the difference is less than this amount then scaling is used, else the file is scanned before loading. Also, if the current size of the symbol file is less than this amount then it will always be scanned. The default value for SymbolGrowth is 5k, although this can be altered to your needs.

When the project is saved, so too are all the usage and sizes of all symbol files. This means that if a symbol file is reloaded without change, then the debugger will know the exact requirements of the file before it is loaded.

If you are curious about actual memory usage, press Ctrl+Shift+I to see a summary.

- Bug causing intermittent gpfs in disassembly pane
- Bug causing intermittent gpfs when showing the callstack and when you have a function with a frame > 4k
- A couple of memory leaks which could decrease system performance when keeping the debugger open for long periods
- **MipsCpu:-**
  - Bug unpicking the stack or forming a callstack. The bug didn't unpick the 'sp' register. This meant that code which was compiled with no frame pointer (any optimisation level) was not unpicked correctly. Note: Optimisations make some functions unpickable if the stack is not needed by the function.

- **History:-**
  - Bug causing intermittent gpf when the debugger added a new item to a history list

### Changes from 13\_1 to 13\_2 - v2.12 to v2.13 beta

- **Debugger:-**
  - Tags for structures were not being found from within overlays when the symbol file had been preceded by Symmunge. The result was tags being shown as <struct> and also prevention of structure expansion.
  - Fixed occasional debugger crash when disassembling off the end of the memory map.
- **MipsCpu:-**
  - lo/hi registers are the wrong way round in a register window
- **SN64 examples:-**
  - The odd floating point registers were not being saved by the exception handler. The object files shipped in examples/n64 have been updated. The latest object files which should be linked with your game image can be obtained separately from the BBS.
- **Driver:-**
  - Fixed slow printf and file serving for the DTL-H2700

## Reporting Bugs

Use the diagnostic tool to generate a report that can automatically be sent to use via email. See 'Check for Errors' in the SN program group on the start menu. If the error you are reporting generates a processor exception, the details from the crash can be copied and pasted into the email too. If you have JIT debugging enabled, then additional stack words will also help to locate the problem.

Send email to [support@snsys.com](mailto:support@snsys.com) or telephone +44 (0)117 9299 733 for priority support.