

# TPF Software News

Volume 1, Issue 2, Fall 1999

© 1999, TPF Software, Inc.

## TPF/GI unveils context-sensitive HTML help

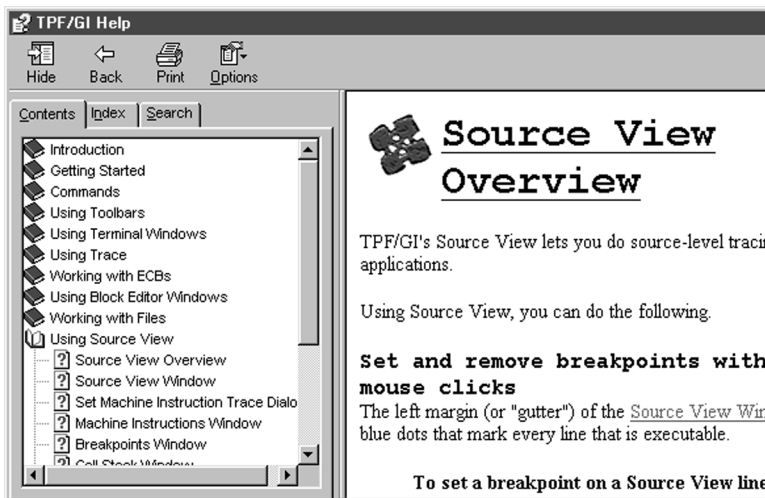


Figure 1: TPF/GI's new context sensitive help window uses Microsoft's HTML help.

In order to put comprehensive help at users' fingertips, TPF Software has completely rewritten the help file for TPF/GI. Over a hundred new help topic entries—including at least one help topic for each TPF/GI window—are now available to answer users' questions.

In addition to window-specific topics, *overview* help topics let users understand the "big picture" surrounding a TPF/GI task, and *step-by-step* topics help users learn a task in detail.

The result is a resource that can help make programmers more self-sufficient as they

develop their application-testing skills.

### Context Sensitive Help

But content is not the only thing that has been improved: accessing the help system is easier because it has been made *context-sensitive*. *Context-sensitive* means that users can press the F1 key at any point and receive information about the window or dialog box that currently has focus.

For windows that were already making use of the F1 key, users can press Ctrl+F1 instead.

---

**With context-sensitive help, TPF/GI users can press the F1 key to learn about the window that is focused.**

---

### The most advanced help format: HTML Help

Last but not least, the TPF/GI help system now uses the latest help format: Microsoft HTML help. The HTML help window, shown in Figure 1, improves on the old WinHelp system by letting users see the table of contents, index, or search window as they view the help topic text.

HTML Help also offers the ability to integrate Web-based help with the local help file, ensuring that users always have the best help available. ❖

### Table of Contents

- 1 TPF/GI unveils context-sensitive HTML help
- 2 TPF/GI Bookmarks get persistent
- 2 Other enhancements
- 3 Plug your own code into TPF/GI with the *Plug-in Tools API*
- 4 A peek into TPF/GI's future
- 4 GI/TERM provides graphical TPF terminal support
- 4 TPFHub launching

# TPF/GI bookmarks get persistent

TPF/GI's bookmark feature lets programmers mark and annotate core blocks, file blocks, program blocks, and other kinds of blocks that they edit with TPF/GI's block editor windows.

For example, suppose a programmer makes a change to a core block and wants to remember what the change was and why she made it. All she has to do is click the "gutter" to the left of the line where the change occurred; this creates a bookmark. Within the bookmark she can type the change she made and the reasons for the change.

block in someone else's program, she can use bookmarks to communicate to the other person what she has seen. She can type in her comments and then make a print out or email a copy of the bookmark file.

To move quickly from one location to another. A bookmark tab appears at the bottom of the editor window every time a programmer adds a bookmark. He can click this tab to jump instantly to the location where he set the bookmark. ❖

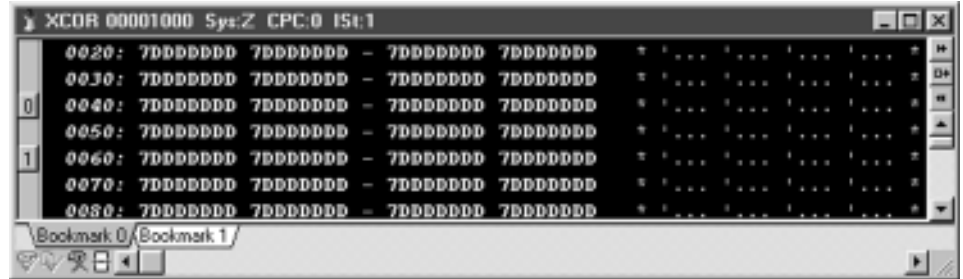


Figure 1: The small buttons in the left gutter mark where bookmarks are set. The tabs at the bottom of the window allow quick navigation between bookmarks. Placing the mouse cursor over a bookmark causes the comments associated with bookmark to appear.

## TPF/GI bookmarks no longer have to be temporary.

The bookmark feature has recently been improved so that bookmarks can be printed, saved to a file, and loaded from a file. Because they no longer have to be temporary, bookmarks have become truly useful.

### Reasons to Use Bookmarks

Here are some uses for bookmarks.

*To document work.* Before making a change to a line, a programmer can place a bookmark beside the line. The current text of the line will be inserted automatically into the bookmark comment field. Once the line is edited, the programmer can add additional comments to the bookmark documenting why the change was made. Finally, the block can be saved or printed out with the bookmarks included in the text.

*To plan and remember.* As a block is examined, the programmer can use bookmarks to make notes to himself so he won't forget things. If he saves his bookmarks to a file, he can reload them—even days later—and resume his work.

*To communicate what needs to be done.* If a programmer is testing or examining a

## More enhancements

Here are a sampling of other new enhancements to TPF/GI and CMSTPF.

### Interactive Trace Enhanced

The interactive trace feature has been enhanced to provide better information for ISO C programmers. Many hard-to-understand codes in the trace output have been replaced by plain language descriptions. Even more impressive, the interactive trace now uses dot notation to reveal which member within a library or DLM is calling a particular macro or function.

### PUT Level 10 Support Ready

CMSTPF now supports the program update tape (PUT) 10 enhancements to TPF. Administrators should contact TPF Software when they are ready to implement PUT 10.

### Collecton Services Support Added

CMSTPF has been enhanced to support TPF collection services (TPFCS).

### Expression Facility Now Works for SABRETalk

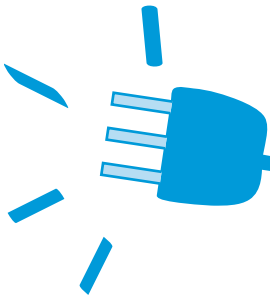
Thanks in great part to support code written by Jeff Longwell, TPF/GI has extended the "assembler" expression facility so that it can be used from SABRE-Talk programs at Worldspan.

The Assembler Expression Facility brings the power of "high-level" language debugging to assembler (and now SABRETalk). It allows programmers to monitor and edit the values of expressions as they test.

### Plug-In Tool API Programmers Get "User Core" Windows

Programmers who are creating plug-in tools (see page 3) now have the power to

*Continued on p. 4*



# Plug your own code into TPF/GI with the *Plug-In Tools API*

**B**ecause all of its customers develop and test applications differently, and because customer needs change and grow, TPF Software provides the Plug-In Tools API. The Plug-In Tools API allows programmers for TPF shops to plug their own code into TPF/GI.

## **What a Plug-In Tool Is**

To a great extent, plug-in tools appear to become part of TPF/GI: they are allowed to create TPF/GI menu items and tool buttons, they are notified of TPF/GI program events, and they can summon TPF/GI windows to appear. Plug-in tools can also create windows of their own.

To create a plug-in tool, programmers use a PC programming environment to write a dynamic link library (DLL). The name of the DLL is then placed in a text file. When TPF/GI starts up, it reads the text file and loads all the DLLs listed there.

---

***DLLs can invoke host execs; working together, DLLs and execs can do almost anything.***

---

## **Using the Plug-In Tools API**

After TPF/GI loads a DLL, it passes the address of the Plug-In API to the DLL. The Plug-In API is a structure that contains a series of pointers to TPF/GI functions. DLLs can call these functions to upload and download files, lock Source View control files, convert buffers between EBCDIC and ASCII, find out the currently active system and CPC, and to do many other things.

The most important API function is the one that allows DLLs to invoke host execs. When the DLL invokes the exec through TPF/GI, the exec runs on the host and passes back information to the DLL on the PC. By writing DLLs and execs that work together, programmers and administrators can accomplish almost anything they want. They can make the entire library system available through TPF/GI, for instance. Or they can automate common tasks, such as loading several programs and locking the control files when one button is clicked.

## **Development Tools**

Plug-in tools can be written in any programming language that can produce a DLL, although the most commonly used development environments for writing plug-in tools are Visual C++ and Borland Delphi.

Programmers who are interested in writing plug-in tools can obtain the necessary Visual

*The TPF/GI Plug-In Tools API allows you to exercise your C++ and EXEC programming skills by building useful tools for yourself and your colleagues.*

*System administrators are using the API to provide library functions, upload and download files, and edit custom data.*

*But the sky and your creativity are the limit.*

C++ header files or Delphi units from the TPFHub web site. Support files for other capable development environments will be created when the need arises.

Documentation, example programs, and a Powerpoint presentation are also available for download.

For more information, please visit [www.tpf-software.com](http://www.tpf-software.com) and click the TPFHub icon. ❖

# A peek into TPF/GI's future

Some of the changes below will arrive with the next version of TPF/GI, others soon after. Either way, continued major enhancements are in store for TPF/GI.

## Action Palette

As a first step toward making it easy to automate common TPF/GI tasks, the next version of TPF/GI will include an "action palette" that allows testers to easily repeat past actions such as adding a Source View control file, setting or removing a breakpoint, and sending in console input.

Both recent actions and "favorite" actions will be remembered by the palette.

## New Plug-In Events, Functions

For Plug-In Tools programmers, the next version of TPF/GI will provide new functions for sending in terminal input and setting and removing Source View breakpoints.

In addition, new API events will notify plug-in tools when users initiate these actions.

## Enhanced C Support

Future versions of TPF/GI will enhance support for displaying C watch variables in their "native" formats and reduce the use of hex-only displays.

## TPF/AR Support

CTFS is being enhanced to include TPF/AR support.

## C++ Support

CMSTPF and TPF/GI are being updated to include C++ support. ❖

## GI/TERM: Graphical Terminal Emulation

GI/TERM is a product from TPF Software that provides graphical terminal emulation for TPF systems. Through a simple, powerful interface, it offers ALC, Local 3270, and SNA 3270 support. GI/TERM provides "emulated" TPF terminal support for both native TPF systems and VPARS.

### Enhances TPF Development Tools

Some development tools in the TPF world currently provide no graphical terminal interface with which to conduct testing. GI/TERM steps in to provide easy graphical access to "emulated" TPF terminals on the PC.

### Enhances Existing Environments

GI/TERM is not just for programmers. Anyone who needs access to a TPF system can use GI/TERM. GI/TERM uses TCP/IP to provide seamless terminal connectivity over the Internet and company intranets.

### TPF Testing Aware.

Because TPF Software has a long history with TPF terminal emulation, it is able to tailor the GI/TERM interface to the concerns that TPF shops have. For example, the recording and playback of messages are integrated directly into the interface of the terminals.

For more information on GI/TERM, please contact TPF Software. ❖

## Enhancements

*Continued from p. 2*

allow users to view and edit custom core windows through the new UCOR API.

UCOR stands for "user core" window. By telling TPF/GI to create a user core window, and by supplying the core address, block size, and overlay panel, API programmers can allow their colleagues to view and edit custom data types.

### Problem Reporting Made Easier

It's easier to report possible bugs in the latest version of TPF/GI. A "More" button in the About box provides detailed TPF/GI version information, and a new menu item under Help allows users to turn on the network trace. ❖

## TPFHub launching

TPF Software is launching TPFHub.com, (<http://www.TPFHub.com>) in order to provide online support for TPF Software products and create an online community for TPF professionals.

The site will feature a message board for TPF Software program support, a message board to discuss TPF issues, answers to frequently asked questions (FAQs), news about TPF Software and TPF, links to TPF-related sites, and downloadable documentation. ❖



## Contact TPF Software

**Web Site** [www.tpf-software.com](http://www.tpf-software.com)

**Email** [info@tpf-software.com](mailto:info@tpf-software.com)

**Telephone** 919-676-5501

**Address** TPF Software, Inc.  
8729 Gleneagles Drive  
Raleigh, NC USA 27613-5419