



IPm Remote Debugging

Abstract:

The GDB program debugger is a valuable IPm development tool, but like any program debugger, it requires the user to have the skills necessary to use these tools. The goal of this document is to provide a knowledgeable developer the necessary information to get a working remote debugging environment (Insight / GDB) running, in as little time possible.

This document assumes that the user has installed the v1.2 SIXNET Linux IADK. Currently, there is no debugging support for the SIXNET Windows IADK.

Insight is a version of GDB that uses Tcl/Tk to implement a graphical user interface. It is a fully integrated GUI, not a separate front-end program. The interface consists of several separate windows, which use standard elements like buttons, scrollbars, entry boxes and such to create a fairly easy to use interface. Each window has a distinct content and purpose, and can be enabled or disabled individually. The windows contain things like the current source file, a disassembly of the current function, text commands (for things that aren't accessible via a button), and so forth.

You may wish to use the standard GDB (command line interface). You can do this by running 'powerpc-linux-gdb' instead of 'powerpc-linux-insight.' However, this document does not cover debugging with GDB.

Note: It is necessary to maintain debugging information in any executable that you are trying to debug. Do not strip (powerpc-linux-strip) your compiled programs.

You can build your programs with the **-g** option to enable debugging.

```
$ powerpc-linux-gcc -g -o hello hello.c
```

Configure the GDBserver:

- Telnet to the IPm station.
- Verify that the program you want to debug is present in the station, and has the appropriate permissions.
- Start the GDBserver.

```
Usage: gdbserver COMM PROG [ARGS ...]  
gdbserver COMM --attach PID
```

COMM may either be a tty device (for serial debugging), or a HOST:PORT to listen for TCP connection.



```
$ gdbserver /dev/ttyS0 ./hello -v
```

The above line starts the server on /dev/ttyS0 (Port B). 'hello' is the program that you wish to debug. The '-v' is a command line option that is passed to the 'hello' program.

```
$ gdbserver 10.1.0.1:2345 ./hello
```

The above line starts a TCP server on the remote station. The server will listen on IP 10.1.0.1 and port 2345. No additional command line arguments are passed to the 'hello' program.

Note: If the program being debugged is not in the default path, it is necessary to specify the path (full or relative) to the program being debugged.

- You may now start the Insight (or GDB) debugger.

Configuring the Insight debugger:

- Start the Insight debugger

```
$ powerpc-linux-insight
```

Note: You may wish to put the debugger in the background by adding an ampersand to the end of the above line.

```
$ powerpc-linux-insight &
```

- Click on the Run Icon (or click on the 'Run' menu and select 'Run').
- Select the appropriate filename for the program you wish to debug.

Note: If this file is not compiled for PowerPC debugging, you will not see any useful debugging information.

- Select the 'Target Settings.'

This is dependent on what the GDBserver is set up for. At a minimum, you will want to select the target connection type (GDBserver/Serial or GDBServer/TCP) and an initial breakpoint.

- Click OK when you are ready to debug.
- Insight should establish a connection with the GDBserver and you should be able to use the debugger as if you are debugging natively.



Installing Insight

The following file contains a pre-built version of Insight:

ppc_insight.tar.gz

It has been built on a standard Red Hat 7.0 installation and should work for most individuals running Red Hat 7.0 or newer. If you have difficulties running the pre-built binaries, it is recommended that you build Insight from scratch.

- Copy the file to /opt/sixnet.

```
$ cp ppc_insight.tar.gz /opt/sixnet/
```

- Uncompress the archive

```
$ tar zxvf ppc_insight.tar.gz
```

- Insight should now be installed in the proper location.

```
$ which powerpc-linux-insight
```

The above should show powerpc-linux-insight exists in /opt/sixnet/bin.

Building Insight & the GDBserver

Depending on the Linux installation, you may need to build Insight from scratch. The following should walk you through the build process. This document does not support missing or invalid dependencies.

- Obtain the latest version of Insight (<http://sources.redhat.com/insight>).
- Uncompress the archive. Depending on what version you download, you will need to do one of the following:

Gzipped tar archive

```
$ tar zxvf insight-5.3.tar.gz
```

or

Bzipped2 tar archive

```
$ bunzip2 insight-5.3.tar.bz2
```

```
$ tar xvf insight-5.3.tar
```



- Make a temporary build directory.

```
$ mkdir build-insight
```

- Change to above mentioned directory.

```
$ cd build-insight
```

- Configure the Insight build.

```
$ ../insight-5.3/configure --target=powerpc-linux --prefix=/opt/sixnet \  
--disable-nls -v
```

Note: The above mentioned configure flags are defined below.

--target	specifies cross target
--prefix	specifies the top-level installation directory
--disable-nls	do not use native language support
-v	verbose configure

- Build Insight

```
$ make all
```

- Install Insight

```
$ make all install
```

You should now have a 'powerpc-linux-insight' and 'powerpc-linux-gdb' in your path. You can test with:

```
$ which powerpc-linux-insight
```

- Change to the GDBserver directory.

```
$ cd ../insight-5.3/gdb/gdbserver
```

- Set **configure** executable.

```
$ chmod 755 configure
```

- Configure the GDBServer.

```
$ ./configure --target=powerpc-linux --prefix=/usr --without-fp -v
```



Note: The above mentioned configure flags are defined below.

--target	specifies cross target
--prefix	specifies the top-level installation directory
--without-fp	unit does not have a floating point unit
-v	verbose configure

- Edit the Makefile.

```
$ vi Makefile
```

Change **CC=gcc** to **CC=powerpc-linux-gcc**

- Build the GDBServer

```
$ make all
```

- Strip the GDBserver

```
$ powerpc-linux-strip gdbserver
```

- FTP **gdbserver** to the IPm station. The default location is /usr/bin.

```
$ ftp 10.1.0.1
```

- Telnet to the IPM station.

```
$ telnet 10.1.0.1
```

- Set gdbserver executable.

```
$ chmod 755 gdbserver
```

- Your debug environment is now installed.

Additional Documentation

- GDB manual pages
- Insight README
- GDBserver README
- Insight Homepage (<http://sources.redhat.com/insight>)
- GDB Homepage (<http://sources.redhat.com/gdb>)