

NAME

MITR(5) -- Itanium(R) Based Systems Only

SYNOPSIS

Mixed mode TRanslator enables (MITR) calling PA-RISC libraries from Integrity binaries running on HP-UX integrity systems.

DESCRIPTION

HP Mixed Mode TRanslator (MITR) allows mixing of PA-RISC and Itanium binaries in an application. Specifically, an Itanium executable can load PA-RISC shared libraries in addition to other Itanium shared libraries. MITR supports transitioning between PA-RISC and Itanium load modules through function calls. The PA-RISC shared libraries are emulated by libraries. MITR only supports dynamic loading of the PA-RISC libraries (using `dlopen(3C)`, `shl_load(3X)`).

Components

MITR consists of following components:

- Enhanced Java Virtual Machine -- `/usr/ccs/bin/mitr/libjvm-1.5[6].so`
- Enhanced ARIES (PA library Emulator) -- `/usr/lib/hpux32/libaries32.so`
- Enhanced dld (dynamic loader) -- `/usr/lib/hpux32/dmm.so`

Tools

In certain cases MITR requires wrapper libraries (which provide information about the types of the arguments and return values) since the runtime architecture conventions are different on PA-RISC and Itanium.

`mitr_driver` An interactive MITR driver script that helps user to:

- Verify if application is eligible to be run in mixed mode.
- Check if wrapper libraries are required.
- Create the necessary wrapper libraries.

It will in turn use the following scripts:

`mitr_verify` Certifies MITR readiness of IPF binaries and PA-RISC libraries.



NOTE: In certain cases MITR requires wrapper libraries (which provide information about the types of the arguments and return values) since the runtime architecture conventions are different on PA-RISC and Itanium.

`ia2pa_wp` Verifies if wrapper libraries are needed for IPF -> PA-RISC calls and creates wrapper libraries.

`pa2ia_wp` Verifies if wrapper libraries are needed for PA-RISC -> IPF calls and creates wrapper libraries.

`chld32` Changes the interpreter path name of the IPF executable to use MITR dld.

`proto-gen` This tool is used by the wrapper library tools to generate prototype information.

All of these MITR tools will be available under `/usr/ccs/bin/mitr` directory.

Running the application under MITR

To run applications with MITR, complete the following steps:

1.
 - a. Verify if application is eligible to run in mixed mode.
 - b. Verify if argument translation can be done transparently.
 - c. Create wrapper libraries if needed.

All of these can be accomplished using the interactive MITR driver script `mitr_driver`. Use `-help` option to know details on available options and how to use them.

```
$mitr_verify PA-RISC_Libname | IPF_Libname | IPF_Executable ...
```

2. Change the path of the dynamic loader to use mixed mode loader by using `chdld32` command.

```
$chdld32 <Itanium Executable>/usr/lib/hpux32/dmm.so
```



NOTE: For a java application, the interpreter path name of IA 64 binary 'java' has to be changed.

```
$chdld32 /opt/java1.[4/5/6]/bin/IA64N/java /usr/lib/hpux32/dmm.so
```

3. Copy mixed mode `libjvm.so` to the correct path.



NOTE: To run java applications in mixed mode (i.e., one or more of the native libraries are PA-RISC), Java 1.4.2.20 (or above) or Java 1.5 (any version) or Java 6 (any version) has to be installed.

This step is only applicable for Java 1.5 and Java 6. For Java 1.4 the installed `libjvm.so` would work fine under MITR.

```
$cp /usr/ccs/bin/mitr/libjvm-1.5[6].so
/opt/java1.5[6]/jre/lib/IA64N/server/libjvm.so
```

4. Run the application.

Debugging support

HP-WDB version 6.0 (and above) can be used to debug core files generated while executing mixed mode applications. For more information, see HP WDB manuals and documentation at <http://hp.com/go/WDB>.

Environment variables

HP_DLD_OPTS

When a library is loaded in mixed mode execution, the dependent libraries are searched in `LD_LIBRARY_PATH`, `SHLIB_PATH`, recorded path and `/usr/lib/hpux32`. But for performance reasons, it is better to load the Itanium versions of the dependent libraries, if available, instead of the PA-RISC versions. To enable this, set the environment variable `_HP_DLDOPTS` to contain the `-pickialib` option. This option ensures that the Itanium versions if available in `/usr/lib/hpux32` are loaded instead of PA-RISC versions of libraries present in `/usr/lib`.

For example

If the dependent library is `/usr/lib/libXt.3`, then `/usr/lib/hpux32/libXt.so` is loaded. The user can specify the exact Itanium library to be loaded by setting the environment variable `_HP_DLDOPTS` to contain the absolute path of the Itanium library. For example, to pick up the Itanium library `/any_path/<libname>.so.3` instead of the PA-RISC library `<libname>.<any_extn>`, set `_HP_DLDOPTS` to contain `-pickialib:/any_path/<libname>.so.3`.

The dynamic loader loads `/any_path/<libname>.so.3` instead of the dependent library `<libname>.<any_extn>`. For all other PA-RISC

dependent libraries that are present in `/usr/lib`, the Itanium versions present in `/usr/lib/hpux32` are loaded.

PX_LIBARIES	The <code>libraries32.so</code> present in <code>/usr/lib/hpux32/</code> is used to emulate PA-RISC libraries in mixed mode execution. To specify an alternate path, set the environment variable <code>PX_LIBARIES</code> to contain the alternate path.
mitr_proto_info	MITR expects that appropriate wrapper libraries be available for successful PA <-> IA call proceedings. In case it is not able to find the appropriate wrapper libraries (or the required prototype information in the wrapper libraries) it emits the detailed warning messages. To enable these messages to be emitted, user has to specify <code>-mitr_proto_info</code> option in the MITR configuration file (<code>.mitrrc</code>). This will help user to create the correct wrapper libraries for their applications.

LIMITATIONS

Certain scenarios that are discussed below may disqualify an application from being executed in mixed mode.

- PA-RISC library has Thread Local Storage (TLS)
MITR can not emulate a PA-RISC library that has TLS. PA-RISC libraries can be built only with static tls since dynamic tls is not supported on PA-RISC. Hence PA-RISC libraries with static tls cannot be dynamically loaded.
- Application has calls to `setjmp/longjmp/getcontext/setcontext/_lwp_getstate/_lwp_setstate`
The MITR solution does not currently support calls to `setjmp/longjmp/getcontext/setcontext`, either from a PA-RISC binary or an Itanium binary.
- PA-RISC library has calls to `sysconf`
The system call `sysconf()` returns `CPU_IA64_ARCHREV_0` when called with `_SC_CPU_VERSION`. PA-RISC libraries that expect to be running on a PA-RISC machine may run incorrectly.
- PA-RISC library has calls to `signal/sigaction`
Operations on/with the PA context data inside the handlers installed using the `signal/sigaction` system calls is not supported.

Many of these scenarios can be detected using the scanner tool provided with this solution.

SEE ALSO

aries(5), *MITR README*