

Getting Started with SAF Test

with an eye on HPI

The short labs here should help you get some hands-on experience with using and creating tests for SAF Test. Hopefully this will enable the tests to more quickly understood, improved, and integrated into the SA Forum certification process.

1. Overview

(15 min)

1.1 Websites

Briefly check out the following websites. All websites shown here are accessible from the SAF Test home page.

- SAF Test home page <http://safetest.sourceforge.net>
- SAF Test sourceforge page <http://sourceforge.net/projects/safetest>
- OpenHPI home page <http://openhpi.sourceforge.net/>
- OpenHPI sourceforge page <http://sourceforge.net/projects/openhpi>

1.2 Mailing List

If you haven't done so already, please sign up on the mailing list for SAF Test. <http://lists.sourceforge.net/lists/listinfo/safetest-devel> Feel free to ask questions. Don't be shy – we're a friendly bunch.

Other related websites:

- SA Forum home page <http://www.saforum.org/home>
- IPMI open source project <http://openipmi.sourceforge.net/>
- SNMP open source project <http://net-snmp.sourceforge.net/>

2. Run SAF Tests on OpenHPI

(30 min)

This section will show you how to use the SAF Test HPI tests on a sample implementation. It will walk through downloading and installing OpenHPI and the SAF Test suite for HPI B.01.01. OpenHPI is a good practice candidate for the certification process because it is open source, easy to install, and already implements all of the HPI B.01.01 specification.

Preparation: We will be downloading and building two projects. Create a directory to put everything in (e.g. ~/saflab)

Clean-up: See the section at the end of this document on how to clean up everything we do here in case you do this on a borrowed system.

2.1 Setup OpenHPI

2.1.1 Download

Go to <http://openhpi.sourceforge.net> Download the 2.0.3 release (currently under “Releases” on the left side, but this could change). You can also try out 2.1.0 if you want to live on the edge.

Save the file in the folder you created above (e.g. ~/saflab/openhpi-2.0.3-1.tar.gz)

2.1.2 Extract, configure, make, and install

The following commands will extract the contents of the downloaded file, configure the code for building on your system, compile the code, and install OpenHPI v2.0.3 on your system. If you are not yet familiar with these commands, you soon will be ☺. You can learn more by typing “man <command>” (e.g. “man tar”) at the command prompt.

Type the following (tab completion is your friend):

```
> cd ~/saflab
> tar -xvzf openhpi-2.0.3-1.tar.gz
> cd openhpi-2.0.3
> configure
> make
> make install (must be root)
```

Almost done... just two more steps.

2.1.3 Create openhpi.conf

Create the file “/etc/openhpi/openhpi.conf”. Use your favorite editor to add the content as shown below.

```
> mkdir /etc/openhpi (must be root)
> vi /etc/openhpi/openhpi.conf (must be root)
```

```
----- Contents of openhpi.conf -----
plugin libdummy
plugin dummy
handler libdummy {
    entity_root = "{SYSTEM_CHASSIS,1}"
    name = "test"
    addr = 0
}
```

2.1.4 Update system library configuration file

Type the following to update the system library configuration file (/etc/ld.so.conf):

```
> ldconfig (must be root)
```

2.2 Setup SAF Test for HPI

2.2.1 Download

Go to <http://safest.sourceforge.net> Download the SAF Test suite for HPI B.01.01 Domain and General sections.

Save the file in the folder created above (e.g. ~/saflab/safest_HPI-B_01_01_0_1_0.tar.gz)

2.2.2 Extract and configure

Type the following (tab completion is your friend):

```
> cd ~/saflab
> tar -xvzf safestest_HPI-B_01_01_0_1_0.tar.gz
> cd safestest
```

Take a look around at the files. You can read about the organization of the directories and meaning of all the files by reading `./doc/Framework_Specification.htm` (also available on the SAF Test website)

Here's a quick summary. Please note the ones in **bold**

File Name	Description
AUTHORS	You, too, can have all the glory
ChangeLog	Modified before each release
COPYING	GPL license
doc	Detailed documentation and coding style
HPI-B.01.01	The beef. Where all the HPI tests are
include	Where the SA Forum specification header files are. There is one for HPI (hpitest_b.h)
log	Useful output from building and running tests
Makefile	
README	Tells us the simple steps to take
report.sh	Outputs the results in XML and TXT formats
run_tests.sh	The script to do the work!

Following the README we have just a few steps to complete.

Edit the LDFLAGS file and uncomment the OpenHPI library (or put in any library you are testing) LDFLAGS tells the tests which implementation(s) to link with.

```
> vi HPI-B.01.01/LDFLAGS

----- Contents of LDFLAGS -----
# Place your library file here
# Format: -l<library name>
# Example: -l<lopenhpi>
-llopenhpi                               ← Uncomment this line (remove the '#')
```

Save the file and exit your editor.

2.2.3 Make the HPI tests

Go back to the SAF Test parent directory and run “make” All the tests will be built.

```
> cd ~/saflab/safptest
> make
```

The log directory contains the output from the build.

2.2.4 Run the HPI B.01.01 tests!

Finally everything is configured. We are ready to run the tests.

```
> run_tests.sh
```

The output on screen is also stored in the file `log/run_log-HPI-B.01.01`. Now you can generate a summary report:

```
> report.sh
```

This creates files with the summary information in two formats: `result.txt` and `result.xml`. Eventually this script (or something much more elaborate) would be used to parse the `run_log-HPI-B.01.01` file generated by `run_tests.sh` to produce the official test output.

3. Contribute a new HPI test to SAF Test

(45 min)

This section will walk through creating accounts on sourceforge, using CVS, analyzing the SA Forum HPI specification and writing an actual test for SAF Test.

3.1 Setup sourceforge account

Go to <http://sourceforge.net> On the left side, click “New user via SSL” Enter the information requested. Don’t worry, sourceforge will not distribute your e-mail, even to people on the mailing list. You must be able to access this e-mail address in order to complete the registration process, so you may want to use an online e-mail address for the purpose of this lab. You can change the e-mail address at any time in the future.

Complete the registration process by replying to the verification e-mail you receive.

Ask Bob Spencer or Crystal Xiong to add you to the developer list.

3.2 Checkout SAF Tests using CVS / ssh

Create a directory to store all the tests (e.g. ~/saflab2) Make sure that it doesn’t have a subdirectory called “safstest” in it. If it does, rename the existing directory.

Check to see that your system has CVS installed:

```
> cvs
```

Setup secure cvs access through ssh:

```
> export CVS_RSH=ssh
```

Checkout the tests from SAF Test

Anonymous

```
> cvs -d:pserver:anonymous@cvs.sourceforge.net:
    /cvsroot/safstest login
<empty password>
> cvs -z3 -d:pserver:anonymous@cvs.sourceforge.net:
    /cvsroot/safstest co -P safstest
```

Secure

```
> export CVS_RSH=ssh
> cvs -z3 -d:ext:<developername>@cvs.sourceforge.net:
    /cvsroot/safstest co -P safstest
```

3.3 Writing a test

For practice we will look at `saHpiSessionOpen`

6.1.1 `saHpiSessionOpen()`

This function opens an HPI session for a given domain and set of security characteristics (future).

Prototype

```
SaErrorT SAHPI_API saHpiSessionOpen (  
    SAHPI_IN SaHpiDomainIdT DomainId,  
    SAHPI_OUT SaHpiSessionIdT *SessionId,  
    SAHPI_IN void *SecurityParams  
);
```

Parameters

DomainId – [in] Domain identifier of the domain to be accessed by the HPI User. A domain identifier of `SAHPI_UNSPECIFIED_DOMAIN_ID` requests that a session be opened to a default domain.

SessionId – [out] Pointer to a location to store an identifier for the newly opened session. This identifier is used for subsequent access to domain resources and events.

SecurityParams – [in] Pointer to security and permissions data structure. This parameter is reserved for future use, and must be set to NULL.

Return Value

`SA_OK` is returned on successful completion; otherwise, an error code is returned. `SA_ERR_HPI_INVALID_DOMAIN` is returned if no domain matching the specified domain identifier exists. `SA_ERR_HPI_INVALID_PARAMS` is returned if:

- A non-null *SecurityParams* pointer is passed.
- The *SessionId* pointer is passed in as NULL.

`SA_ERR_HPI_OUT_OF_SPACE` is returned if no more sessions can be opened.

3.3.1 Conformance Criteria

The tests must provide good enough coverage to confidently assert whether the implementation under test follows the specification line-by-line.

- Each method in the specification must be covered.
- For each method, each line describing the parameters, their values, and the expected return value should be tested, including:
 - a thorough number of possible parameters for those methods,
 - multiple valid and invalid parameter values

- boundary conditions
- error conditions
- validation of the out parameters

3.3.2 Review test case for saHpiSessionOpen()

Go to the directory ./HPI-B.01.01/src/session/saHpiSessionOpen
Open the file 1.c

```
> vi 1.c
```

3.3.3 SAF Test structure

The following items must be included with each test:

- Name corresponding to section in assertion.xml
- Description
- Specification line #
- Expected return value
- Expected value of [OUT] parameters

See http://safest.sourceforge.net/test_review.php to understand how these pieces will be used for reviewing the tests.

3.3.4 Coding Guidelines

To help with coding style consistency, a brief coding guideline has been created. Please check this before submitting your first tests.

```
> vi ~/saflab/safest/doc/coding_style.txt
```