

Verifying the Installation

Use online commands to verify the SMARTS HTTP server installation.

This chapter covers the following topics:

- Verify Operation of the Servers
- Verify the SMARTS HTTP Server Installation
- Troubleshooting

Verify Operation of the Servers

The sample programs and HTML files described in this section are used to verify the operation of the HTTP server.

Sample Programs

The following table lists the sample programs used for each verification stage and the output they produce. All programs are also provided in source format.

Program	Sample . . .	Output
HCANSAMP	C program that takes a content value as CGI input and sends it back to the user.	The value sent as the content value.
HOANSAMP	COBOL CGI program that shows the uses of HAANUPR.	The value sent as the content value.
HOANCONV	COBOL CGI program that makes use of conversational mode.	The value sent as the content value and the previous value.
HPANSAMP	PL/I CGI program that shows the uses of HAANUPR.	The value sent as the content value.

Sample HTML Files

The following table lists the sample HTML documents used during verification processing where:

ip-addr	is the IP address of the TCP/IP subsystem where the HTTP server is running
port	is the port number specified in your HTTP server configuration

Note:

You must change the URL to reflect your version of the dataset name. For example, if HTPvrs.TEST.USER.SRCE is your name for the dataset referred to in this document as HTPvrs.USER.SRCE, the first URL in the following table would be:

<http://ip-addr:port/htpvrs/test/user/srce/PXANCONF>

http://ip-addr:port/	Description
httpvrs/user/src/PXANCONF	a basic test of the HTML functions; displays the PXANCONF configuration file in text format on a WWW browser.
httpvrs/src/hhancget.htm	sample HTML form that drives the sample C CGI program HCANSAMP for HTTP GET processing.
httpvrs/src/hhancobt.htm	sample HTML form that drives the sample COBOL CGI program HOANSAMP for HTTP GET processing.
httpvrs/src/hhanpl1t.htm	sample HTML form that drives the sample PL/1 CGI program HPANSAMP for HTTP GET processing.
httpvrs/src/hhannatt.htm	sample HTML form that drives the sample Natural CGI program HNANSAMP for HTTP GET processing. Note: Natural CGI processing capability must be installed before this HTML can be run.
cgi/hoanconv	runs the sample COBOL conversation CGI program HOANCONV. HOANCONV 'ages' the last two pieces of data as it is entered, illustrating how user context can be maintained over multiple HTML outputs. Enter 'stop' to terminate the conversation.
cgi/<program>	runs <program> which may be any sample program from the previous section Sample Programs; displays program output on a WWW browser.

Verify the SMARTS HTTP Server Installation

The HTTP server installation is verified by running the HTTP server and by accepting and processing HTTP requests from a standard browser.

Prepare the Sample Programs

Compile the C sample HCANSAMP and link it using the SMARTS headers and stubs. See the SMARTS Installation and Operations Manual for details of example jobs.

Refer to the chapter Installing Natural CGI before attempting to run the Natural CGI test programs.

Start the HTTP Server

Start up the SMARTS server environment and ensure the HTTP server starts correctly.

To verify that the HTTP server is operating correctly, run each of the tests described in Sample HTML Files.

Troubleshooting

The HTTP Server Initialization Fails

If the SMARTS environment initialization fails, the HTTP server initialization will also fail.

If the SMARTS environment initialization fails, its configuration is probably invalid. See the SMARTS environment installation and verification information in the SMARTS Installation and Operations Manual.

The SMARTS Environment Initializes, but the HTTP Server Initialization Fails

The HTTP server configuration is probably invalid.

- Check for messages during the HTTP server initialization process that may identify the problem.
- Check in particular for sockets errors that occur if the specified port is already in use by another application in your system.

All SMARTS Components Initialize, but Access Attempts Fail

If SMARTS components issue messages about successful completion of tasks, the problem is probably in the request being issued or the port number being used. Ensure that

- the IP address or DNS name used to identify the target host actually identifies the TCP/IP stack with which SMARTS is communicating. More than one IP address is possible on a mainframe system: if you use the wrong one, your request will time out or be rejected.
- you can connect to the IP addresses by pinging that node. If the ping fails, you have no physical connection to that host.
- the port specified is the one the HTTP server is configured to work with. The default is port 8080.