

Interactive System Productivity Facility (ISPF)



TSO/ISPF client gateway (z/OS Version 1 Release 8.0)

Interactive System Productivity Facility (ISPF)



TSO/ISPF client gateway (z/OS Version 1 Release 8.0)

Contents

About this document v

TSO/ISPF client gateway 1

Gateway description 1

Installing and customizing the gateway 1

 Installation considerations 1

 Installing the gateway 2

 Customizing the gateway 3

 Customizing the HTTP server 4

 Verifying the HTTP server customization 5

 Customizing Remote Systems Explorer 8

 Configuration for other environments 9

Using the gateway 9

 Passing requests to the gateway 10

 Receiving output from the gateway 13

 Return and reason codes from the gateway . . . 17

 Using the native API 18

About this document

This document provides details of documentation changes associated with APAR OA24482 shipped for Interactive System Productivity Facility (ISPF), Version 1 Release 8.0.

The documentation changes apply to ISPF *Planning and customizing* V1 R8 (GC34-4814-05).

If you have applied the relative PTF to your release of ISPF, you should read this document in conjunction with the *Planning and customizing* document.

TSO/ISPF client gateway

ISPF provides an interface or gateway for client applications to connect to the z/OS host and run TSO and ISPF commands. This section describes how the gateway works, how to install and customize the interface, and how client applications can use the interface.

Gateway description

The gateway runs within z/OS UNIX and is invoked upon a request for a TSO or ISPF command from the client application. To run a requested command, the gateway establishes a TSO address space and, if the request is for an ISPF command, data sets are allocated and an ISPF session is also initialized. After completion of the command, the results are returned to the client.

If requested by the client, the gateway maintains the state of the TSO/ISPF session for any subsequent client function requests for the user. Otherwise, the interface terminates the TSO/ISPF session. The advantages of maintaining a user's TSO/ISPF session on z/OS are in bypassing the overhead of having to establish a new session for every TSO or ISPF command request and in adding an inherent ability to maintain state data between calls from the client.

The gateway does not provide the data transport between the client and z/OS host. The gateway is designed to interface with an authenticated connection between the client and z/OS host such as the HTTP protocol (web-based services), direct TCP/IP socket connections, or any other means through which data and commands can be exchanged between the client and z/OS host.

Note: ISPF commands invoked by means of the gateway run in an ISPF batch environment.

Installing and customizing the gateway

This section describes installing and customizing the gateway.

Installation considerations

Consider these points before you configure your system:

- A RACF OMVS segment (or equivalent) that specifies a valid non-zero uid, home directory, and shell command must be defined for each user of the interface.
- Set MAXPROCUSER in BPXPRMxx parmlib member to a minimum of 50. This can be checked and set dynamically (until the next IPL) with the following commands (as described in *z/OS MVS System Commands*, SA22-7627):

```
DISPLAY OMVS,0
SETOMVS MAXPROCUSER=50.
```

Setting a value that is too low can cause the interface to fail.

- Table 1 describes the load modules that comprise the gateway:

Table 1. Load modules comprising the gateway

Name	Library	Description
ISPZINT	ISP.SISPLPA	Gateway initialization - manages spawned TSO sessions, routes command requests to these sessions.
ISPZTSO	SYS1.LINKLIB	TSO initialization - attaches a TSO session to run a command. ISPZTSO must run APF-authorized.
ISPZCNT	ISP.SISPLOAD	ISPF initialization - allocates data sets and starts an ISPF session.
ISPZCMD	ISP.SISPLOAD	ISPF command interface - invokes an ISPF command and, if required, keeps the ISPF session active.

These load modules must be available to run under the server used to communicate between the client and z/OS host.

A PROGRAM class profile that identifies ISPZINT as being a controlled program should be defined using the command:

```
DEFINE PROGRAM ISPZINT ADDMEM('ISP.SISPLPA'//NOPADCHK) UACC(READ)
```

(Refer to your security server's documentation for details.) Failure to define ISPZINT as being controlled might result in messages, such as CSV042I or ICH422I, being issued.

- Table 2 describes the gateway files installed into the z/OS UNIX file system:

Table 2. Load modules comprising the gateway

Name	Description
/usr/lpp/ispf/bin/ISPZINT	A stub file with the sticky bit set on to enable invocation of the gateway initialization load module ISPZINT.
/usr/lpp/ispf/bin/ISPZTSO	A stub file with the sticky bit set on to enable invocation of the TSO initialization load module ISPZTSO.
/usr/lpp/ispf/bin/ISPZXENV	A REXX routine called by the gateway XML API routine, ISPZXML, to set up environment variables.
/usr/lpp/ispf/bin/ISPZXML	A REXX routine that is the XML API for the gateway.

Installing the gateway

To install the gateway, customize and run the JCL in the member ISPZINS1 in the ISPF samples data set ISP.SISPSAMP. Customize the JCL according to the instructions in the member.

This job performs these tasks:

- Creates CONFIG and WORKAREA directories in the z/OS UNIX file system at the directory you specify.
- Copies the sample ISPF configuration table from member ISPZISPC in the samples data set ISP.SISPSAMP to file ISPF.conf in the CONFIG directory. This file requires customization.

The recommended base directory for the configuration files is /etc/ispf. The part of the directory up to ISPF must exist before running this job.

You must have read and write access to the WORKAREA directory /var/ispf/WORKAREA. The WORKAREA is used for the transfer of files. Temporary directories of the format /var/ispf/WORKAREA/userid/* are created during use of the interface.

Note: Some temporary session files might be created in the /tmp directory. Ensure all users have write access to the /tmp directory.

The interface removes any temporary files it creates in the WORKAREA directory. However, temporary output is sometimes left over, for example, if there is a communication error while processing. For this reason, it is recommended that you clear out the WORKAREA directory from time to time. To do this, use these commands in OMVS:

```
cd /var/ispf/WORKAREA
rm -r *
```

Where /var/ispf/WORKAREA depends on where you create the WORKAREA directory.

Customizing the gateway

You must customize the ISPF configuration file ISPF.conf to host site requirements for ISPF data set allocation. This file is stored, by default, in directory /etc/ispf. The provided sample ISPF.conf has instructions to complete customization so your user site can:

- Include the minimum ISPF data set allocations for the gateway to operate. This means allocating the minimum ISPF data sets required to initialize an ISPF session. In the sample provided, these are specified as the isp.sisp* data sets. You might need to change these for your site-specified data set names.
- Add additional DDNAME file allocations or concatenate additional ISPF data sets.
- Launch a customer-defined allocation executable (exec) to provide further data set allocation by user ID. A sample exec is provided in member ISPZISP2 in the ISPF samples data set ISP.SISPSAMP.

The allocations for each of the ISPF DDs must be specified on a single line with each data set separated by a comma. The maximum length of the string defining the data sets for a DD is 255 characters. If the data sets you want to define for a DD exceed this length, use the customer defined allocation exec to allocate the DD.

Comment lines can be added by beginning the line with an asterisk (*). Here is a sample ISPF.conf:

```
* REQUIRED:
* Below is the minimum requirements for ISPF allocation.
* Change the default ISPF data set names below to match your
* host site.
* Add additional dsn concatenations on same line and separate
* by comma.
* Order of data sets listed is search order in concatenation.
*
sysproc=ISP.SISPCLIB
sysexec=SYS1.LOCAL.EXEC,ISP.SISPEXEC
isplib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISPLOAD,SYS1.LOCAL.LOAD
```

The `ISPF_timeout` option in the ISPF configuration file can be used to specify a time out value for reusable ("stateful") ISPF sessions. The time out value specifies the number of seconds a user's ISPF session can remain idle between service call requests. If the idle time exceeds the time out value, the session is terminated and the next service request for the user results in a new ISPF session being established. The default reusable ISPF session time out value is 900 seconds (15 minutes).

Here is an example of setting a time out value of 300 seconds (5 minutes):

```
ISPF_timeout = 300
```

Note: If you have the PDSMAN product installed, it must be disabled for use with the gateway by including the record in the `ISPF.conf` file:

```
ezyoff=nullfile
```

Customizing the HTTP server

If you plan to use the HTTP server to invoke the gateway, these changes must be made to the HTTP environment and configuration files:

- The HTTP environment file, `httpd.env`, must be updated to include directives that identify the workarea and configuration directories for the gateway. The `CGI_ISPWORK` directive is used to define the path for the `WORKAREA` directory used by the gateway. The `CGI_ISPCONF` directive is used to define the path for the `CONFIG` directory where the ISPF configuration file `ISPF.conf` is stored. Shown here are examples of the `CGI_ISPWORK` and `CGI_ISPCONF` directives specifying the default paths for the `WORKAREA` and `CONFIG` directories:

```
CGI_ISPWORK=/var/ispf
CGI_ISPCONF=/etc/ispf
```

- These `pass` and `exec` directives must be added into the HTTP configuration file `httpd.conf`:

```
Pass          /ISPZIVP.html /usr/lpp/ispf/bin/ISPZIVP.html
Exec          /ISPZINT      /usr/lpp/ispf/bin/ISPZINT
Exec          /ISPZXML     /usr/lpp/ispf/bin/ISPZXML
Exec          /ISPZIVP.cgi /usr/lpp/ispf/bin/ISPZIVP.cgi
```

Note: The path specified for these directives must be the path of the directory where the gateway was installed.

If the gateway modules (see Table 1 on page 2) are not in the `LINKLIST`, then update the HTTP server `STEPLIB DD` to include the load library data sets containing these modules.

Note: To use the gateway, the client is required to supply a valid z/OS user ID and password.

For additional information about configuring IBM HTTP Web servers, review these IBM manuals:

- *HTTP Server Planning, Installing and Using*, (SC31-8690)
- *OS/390 e-business Infrastructure: IBM HTTP Server V5.1 for OS/390*, (SG24-5603-00)

Verifying the HTTP server customization

This Installation Verification Process (IVP) applies if you have configured the HTTP server to invoke the gateway. The HTTP server must be running and the IVP pass/exec directives configured in the httpd.conf file for successful verification processing.

From a browser, type the location URL address:

`http://hostname:portnumber/ISPZIVP.html`

where:


hostname

Is the TCP/IP host name the HTTP server is running on.

port number

Is the port used in the job and the httpd.conf file (default port 80).

If the HTTP server is running, you are prompted for a valid TSO user ID and password for the system the Web server is started on:



Connect to pthisd1.au.ibm.com

System_Logon

User name: [User icon]

Password: [Text box]

Remember my password

OK Cancel

After you have entered your TSO user ID and password, the browser initially displays the HTTP welcome screen:



Figure 1. HTTP welcome screen

If you fail to connect, then check that:

- The HTTP server has successfully initialized.
- The z/OS UNIX System Services file system mount point containing the TSO/ISPF Client Gateway installation is mounted.
- The hostname:port are correct (try pinging the Hostname).
- There are no firewall restrictions.
- The PASS directive in the `httpd.conf` file is set correctly:

```
Pass          /ISPZIVP.html /usr/lpp/ispf/bin/ISPZIVP.html
```

After you receive the welcome screen, continue with the IVP, which checks and validates your installation and customization process, by taking these steps:

1. Enter the command `tso lista` in the ISPF command field.
2. Select the **Run installation verification** checkbox.
3. Click on the **Submit** push-button.

The sample screen shown in Figure 2 on page 7 and Figure 3 on page 8 give an example of the expected validation responses:

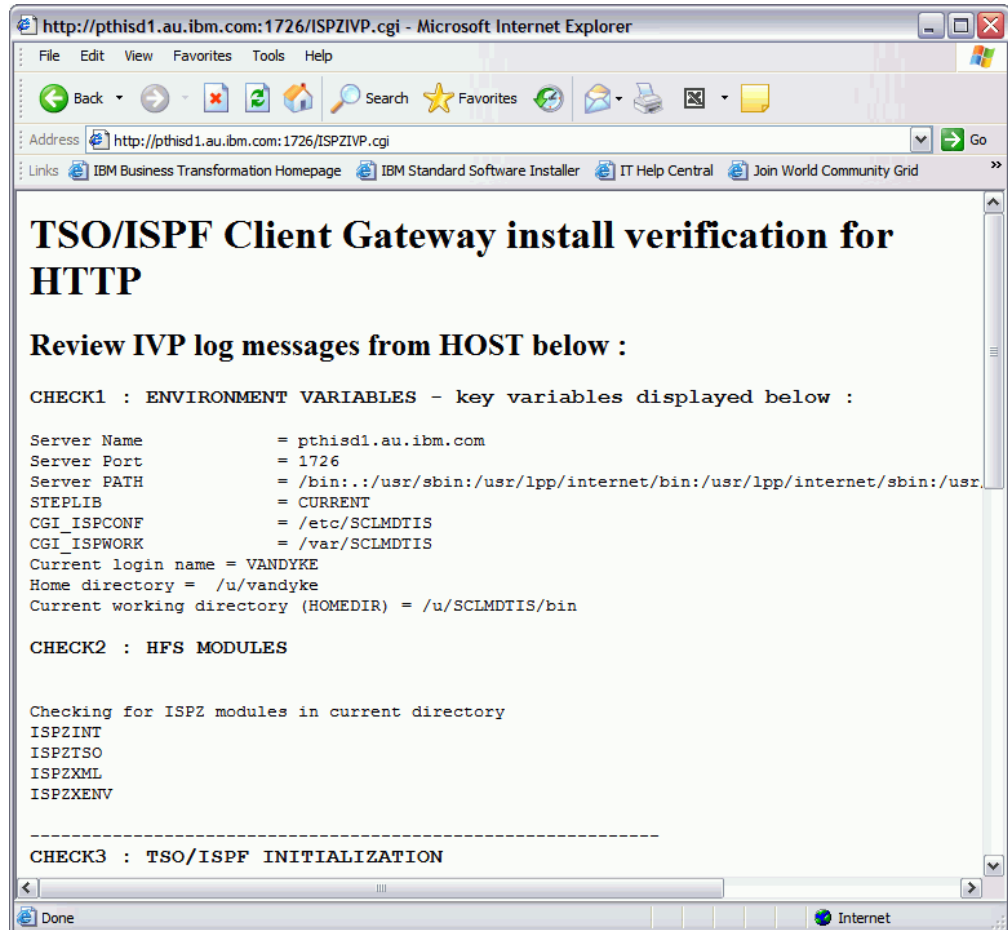


Figure 2. Example of expected validation response

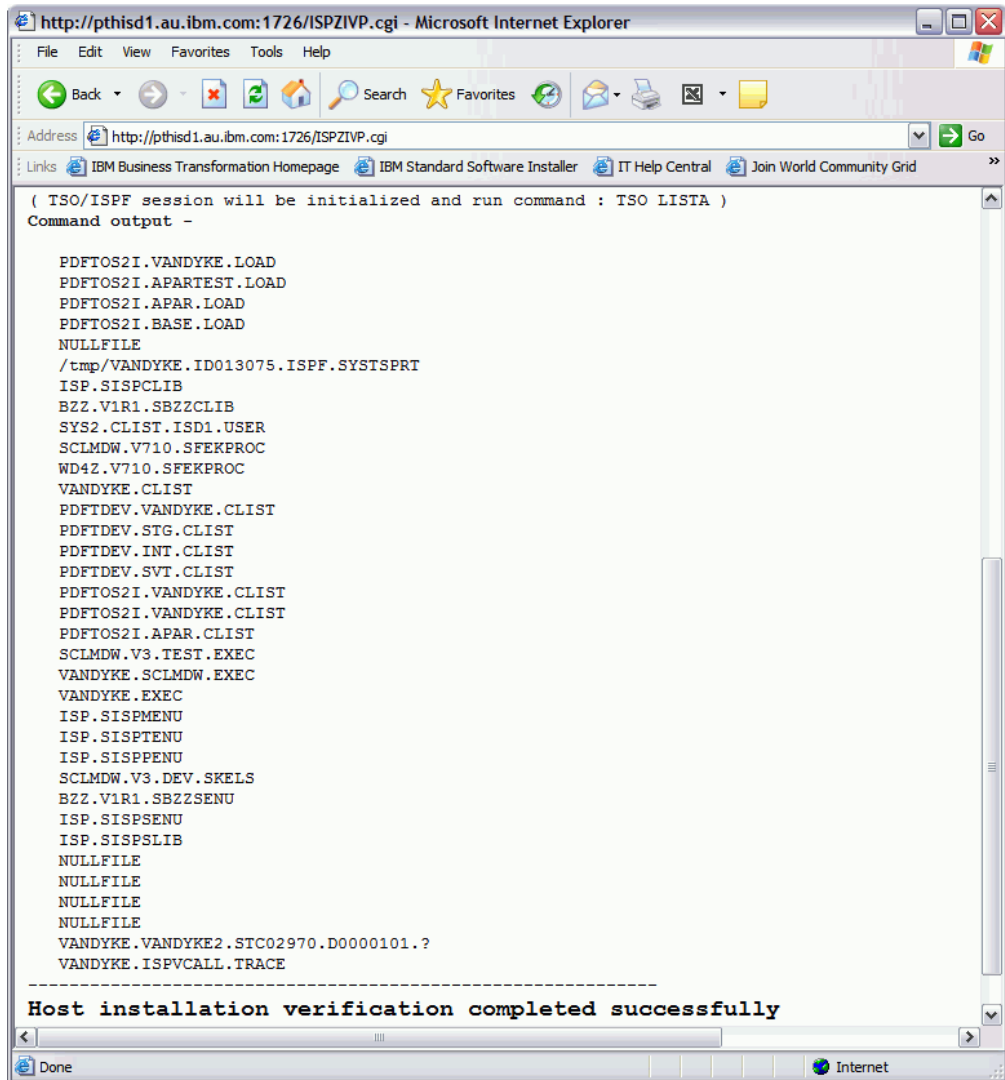


Figure 3. Example of expected validation response (contd.)

Customizing Remote Systems Explorer

If you plan to use IBM Rational Developer for System z's Remote Systems Explorer (RSE) to invoke the gateway, the RSE environment file must be customized. The environment file for your RSE connection is `rsed.envvars` which, by default, is in `/etc/rdz`.

These environment variables used by the gateway should already have been defined in `rsed.envvars` during base RSE customization:

```

_CMDSERV_BASE_HOME
_CMDSERV_CONF_HOME
_CMDSERV_WORK_HOME
  
```

These following changes are required to the `rsed.envvars` file:

- If the gateway modules (see Table 1 on page 2) are not in the `LINKLIST`, then update the `STEPLIB` environment variable value to include the load library data sets containing these modules.

For additional information about installing and configuring RSE, review these IBM documents:

- *Program Directory for IBM Rational Developer for System z (GI11-8298-00)*
- *IBM Rational Developer for System z Host Configuration Guide (SC31-6930-02)*

Configuration for other environments

If you plan to invoke the gateway by means other than through the HTTP server of RSE (for example, from the OMVS shell), configuration is required to ensure values are set for environment variables used by the gateway. This requires the environment set-up routine, `ISPZXENV`, to be modified to set values for the environment variables shown in Table 3.

Table 3. Environment variables to be set by the environment set-up routine, `ISPZXENV`

Environment variable	Description
STEPLIB	Specifies the STEPLIB data sets from where MVS load modules are run. This environment variable must be set to the names of the load library data sets containing the gateway modules (see Table 1 on page 2) if these modules are not in the LINKLIST.
CGI_ISPCONF	Specifies the CONFIG directory path where the configuration files are stored.
CGI_ISPWORK	Specifies the WORKAREA directory path that is used for temporary files.

A sample of this REXX routine is provided in member `ISPZXENV` in the ISPF samples data set `ISP.SISPSAMP`. These lines in `ISPZXENV` may require modification:

```
STEPLIB = 'ISP.SISPLPA:ISP.SISLOAD'
```

The `STEPLIB` variable should specify the data sets containing the load modules for the TSO/ISPF client gateway (see Table 1 on page 2). If these load modules reside in the LINKLIST, then the `STEPLIB` variable should be set to a null value (that is, '').

```
CGI_ISPCONF = '/etc/ispf'
```

The `CGI_ISPCONF` variable specifies the HOME directory path where the configuration files reside for the gateway. For more information on the CONFIG directory, see “Installing the gateway” on page 2.

```
CGI_ISPWORK = '/var/ispf'
```

The `CGI_ISPWORK` variable specifies the pathname of the directory where work files for the gateway are stored. For more information on the work directory, see “Installing the gateway” on page 2.

Replace the version of `ISPZXENV` in `usr/lpp/ispf/bin` with your modified version of `ISPZXENV`.

Using the gateway

This section describes how to provide TSO/ISPF service and command requests to the gateway and the format of the output returned by the gateway.

Passing requests to the gateway

TSO/ISPF service and command requests are passed to the gateway in XML format. The XML schema shown here, which is supplied in member ISPZXSDI in the ISPF samples data set ISP.SISPSAMP, describes the format and can be used to validate the XML for a service request:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="ISPF-INPUT">
    <xs:complexType>
      <xs:all>

        <xs:element name="SERVICE-REQUEST">
          <xs:complexType>
            <xs:all>

              <xs:element name="service">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <!-- Specifies native TSO or ISPF service call -->
                    <xs:enumeration value="ISPF"/>
                    <xs:enumeration value="TSO"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>

              <xs:element name="session" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <!-- Default NONE : Session terminates after service call -->
                    <xs:enumeration value="NONE"/>
                    <!-- Reuseable ISPF session stays active between calls -->
                    <xs:enumeration value="REUSE"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>

              <!-- Use existing ISPF profile in call -->
              <xs:element name="ispprof" type="xs:string" minOccurs="0"/>

              <!-- Free form TSO/ISPF command -->
              <xs:element name="command" type="xs:string"/>

            </xs:all>
          </xs:complexType>
        </xs:element>

      </xs:all>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

This is an example of the XML to request a TSO LISTCAT command:

```
<?xml version="1.0"?>
<ISPF-INPUT>
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ispf.xsd"
  <SERVICE-REQUEST>
    <service>TSO</service>
    <session>NONE</session>
    <command>LISTC ENT('SYS1.LINKLIB')</command>
  </SERVICE-REQUEST>
</ISPF-INPUT>
```

This is an example of the XML to request to run REXX program DINFO in a "reusable" ISPF session which remains active to run subsequent commands for this user:

```
<?xml version="1.0"?>
<ISPF-INPUT>
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ispf.xsd">
<SERVICE-REQUEST>
<service>ISPF/service>
<session>REUSE</session>
<command>TSO DINFO sys1.linklib</command>
<ispprof>USER.ISPPROF</ispprof>
</SERVICE-REQUEST>
</ISPF-INPUT>
```

This shows the REXX program DINFO:

```
/* REXX */
parse upper arg dsn .
address "ISPEXEC" "dsinfo dataset('dsn')"
if rc = 0 then do
  say 'Volume           = ' zdsvol
  say 'Primary allocation = ' strip(zds1ex) zdsspc
  say 'Seconday allocation = ' strip(zds2ex) zds2spc
end
else do
  say 'DSINFO rc = ' rc
  say 'ZERRMSG   = ' zerrmsg
end
exit rc
```

The Java program shown here, which is supplied in member ISPZXJAV in the ISPF samples data set ISP.SISPSAMP, shows an example of invoking the interface through a HTTP server and passing XML to request running the TSO PROFILE command:

```
package com.ibm.ispfcall;

import java.io.*;
import java.net.*;
import java.util.*;

public class XmlIspf {

  public static void main(String[] args) {

    try {

      BufferedReader in = new BufferedReader(new FileReader("C:\\\\logon.txt"));
      String logon = in.readLine();
      in.close();

      Date d = new Date() ;
      System.out.println("START Transfer DATE/TIME is : " + d.toString() );

      // URL details for CGI POST
      URL url = new URL("http", "SITE.COM", 1234, "/ISPZXML");
      HttpURLConnection con = (HttpURLConnection) url.openConnection();

      con.setUseCaches(false);
      con.setDoInput(true);
      con.setDoOutput(true);
      con.setRequestMethod("POST");
      con.setRequestProperty( "Authorization", "Basic "
        + Base64Encoder.encode( logon ));

    }

  }

}
```

```

System.out.println("At url.openConnection.. ");

// POST CGI routines
doPut(url, con, "POST", "");
doGet(url, con);

Date c = new Date() ;
System.out.println("TOTAL Completion DATE/TIME is :"+ c.toString() );

}
catch (IOException exception)
{
System.out.println("Error: " + exception);
}
}

public static void doPut(URL url, HttpURLConnection con, String ISPF_FUNC,
String fileIn) throws IOException
{
if (ISPF_FUNC.equals("POST"))
{
PrintWriter out = new PrintWriter(con.getOutputStream());
// Below is a sample inline XML input for an ISPF service request
// This could alternatively be read from an external file
out.println( "<?xml version='1.0'?">" );
out.println( "<ISPF-INPUT>" );
out.println( "xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'" );
out.println( "xsi:noNamespaceSchemaLocation='ispf.xsd'" );
out.println( "<SERVICE-REQUEST>" );
out.println( "<service>ISPF</service>" );
out.println( "<session>NONE</session>" );
out.println( "<command>TSO PROFILE</command>" );
out.println( "<ispprof>USERID.ISPPROF</ispprof>" );
out.println( "</SERVICE-REQUEST>" );
out.println( "</ISPF-INPUT>" );
out.flush();
out.close();
}
}

public static void doGet(URL url, HttpURLConnection con) throws IOException
{
BufferedReader in;
try
{
System.out.println("About to accept response from Server");
in = new BufferedReader(new InputStreamReader(con.getInputStream()));
System.out.println("Response from Server received");
}
catch (FileNotFoundException exception)
{
InputStream err = ((HttpURLConnection)con).getErrorStream();
if (err == null) throw exception ;
in = new BufferedReader(new InputStreamReader(err));
}

String line;
while ((line = in.readLine()) != null)
System.out.println(line);

in.close();
}
}

```

Receiving output from the gateway

The output from a service request is returned by the gateway to the client in XML format. The XML schema shown here, which is supplied in member ISPZXSDO in the ISPF samples data set ISP.SISPSAMP, describes the format and can be used to process the XML returned for a service request:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="ISPF-OUTPUT">
    <xs:complexType>
      <xs:all>

        <xs:element name="SERVICE-REQUEST"/>

        <xs:element name="SERVICE-RESPONSE">
          <xs:complexType>
            <xs:all>
              <xs:element name="ISPF-COMMAND"/>
              <xs:element name="RETURN-CODE"/>
              <xs:element name="ISPF"/>
            </xs:all>
          </xs:complexType>
        </xs:element>

        <xs:element name="OPERATIONS-LOG"/>

      </xs:all>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

The following shows the XML returned for the TSO LISTCAT command above:

```
<?xml version="1.0"?>
<ISPF-OUTPUT>
  <SERVICE-REQUEST>
    <service>TSO</service>
    <session>NONE</session>
    <command>LISTC ENT('SYS1.LINKLIB')</command>
  </SERVICE-REQUEST>
  <SERVICE-RESPONSE>
    <ISPF-COMMAND>

      </ISPF-COMMAND>
      <![CDATA[
        NONVSAM ----- SYS1.LINKLIB
          IN-CAT --- CATALOG.MASTER.SYSPLEXD
        READY
        END
      ]]>
    </TSO>
  </SERVICE-RESPONSE>
  <OPERATIONS-LOG>
    <![CDATA[
      Content-type: text/plain

      Entering ISPZINT (Service initialization)
      About to read from fileno(stdin) = 0
      Data read from STDIN is TSO LISTC ENT('SYS1.LINKLIB')
      EPOCH secs = 1202109526
      Local Date & time: Mon Feb  4 02:18:46 2008
      Hour: 2 Min: 18 Sec 46
      Function ID timestamp = ID008326
      Environment variables:
```

```

0 QUERY_STRING=
1 CONTENT_TYPE=application/x-www-form-urlencoded
2 PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products/java142/J1.4/
3 AUTH_TYPE=Basic
4 DOCUMENT_URI=/ISPXML
5 SHELL=/bin/sh
6 HTTPS=OFF
7 HTTP_ACCEPT=text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
8 HTTP_USER_AGENT=Java1.3.1
9 SERVER_PORT=1726
10 RULE_FILE=//DD:CONF
11 GATEWAY_INTERFACE=CGI/1.1
12 PATH_INFO=
13 CONTENT_LENGTH=274
14 _CEE_RUNOPTS=ENVAR("_CEE_ENVFILE=//DD:ENV")
15 _BPX_SPAWN_SCRIPT=YES
16 REFERER_URL=
17 _=./ISPZINT
18 CLASSPATH=./usr/lpp/internet/server_root/CAServlet
19 STEPLIB=CURRENT
20 REQUEST_METHOD=POST
21 REMOTE_ADDR=9.190.236.239
22 LANG=C
23 LIBPATH=/bin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin
24 REMOTE_USER=USERNME
25 SERVER_ADDR=192.168.123.11
26 FSCP=IBM-1047
27 PATH_TRANSLATED=
28 HTTP_CONNECTION=keep-alive
29 SERVER_TOKEN=1
30 HTTP_HOST=PTHISD1.AU.IBM.COM
31 _BPX_SHAREAS=YES
32 CGI_ISPCONF=/etc/ispf
33 SERVER_SOFTWARE=IBM HTTP Server/V5R3M0
34 REPORTBITS=77
35 DOCUMENT_ROOT=usr/lpp/ispf/bin
36 NETCP=ISO8859-1
37 CGI_ISPWORK=/var/ispf
38 COUNTERDIR=NULL
39 LC_ALL=en_US.IBM-1047
40 CGI_DTCONF=/etc/ispf
41 _BPX_USERID=USERNME
42 SERVER_PROTOCOL=HTTP/1.1
43 JAVA_HOME=/usr/lpp/products/java142/J1.4/
44 HTTPS_KEYSIZE=
45 TZ=EST5EDT
46 _CEE_ENVFILE=//DD:ENV
47 _BPX_BATCH_SPAWN=SPAWN
48 SCRIPT_NAME=/ISPXML
49 NLS_PATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/internet/%L/%N:/usr/lib/nls/msg/En_US.IBM-1047/%N
50 CGI_DTWORK=/var/ispf
51 DOCUMENT_NAME=usr/lpp/ispf/bin/ISPXML
52 SERVER_NAME=PTHISD1.AU.IBM.COM
Number of environment variables is 53
Protocol = HTTP
FSCP = IBM-1047
NETCP = ISO8859-1
CGI_ISPCONF = /etc/ispf
CGI_ISPWORK = /var/ispf
CGI_TRANTABLE =
Server PATH = /bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products/java
About to spawn task for ISPZTSO
Parameters passed to ISPZTSO - LISTC ENT('SYS1.LINKLIB')
Return code from ISPZTSO is 0
About to open /tmp/USERNME.ID008326.SYSTSPRT
*** XML-NOTE *** Reference tagged SERVICE-RESPONSE

```

```

    OUT.84
  ]]>
</OPERATIONS-LOG>
<ISPF-OUTPUT>

```

Shown here is the XML returned from running the DINFO REXX program in a "reusable" ISPF session (see above):

```

<?xml version="1.0"?>
<ISPF-OUTPUT>
  <SERVICE-REQUEST>
    <service>ISPF</service>
    <session>REUSE</session>
    <command>TSO DINFO sys1.linklib</command>
    <ispprof>USERME.TEST.ISPPROF</ispprof>
  </SERVICE-REQUEST>
  <SERVICE-RESPONSE>
    <ISPF-COMMAND>
      SELECT CMD(DINFO sys1.linklib) NEST
    </ISPF-COMMAND>
    <RETURN-CODE>0</RETURN-CODE>
    <ISPF>
      <![CDATA[
        Volume           = $$SR86
        Primary allocation = 2398 BLOCK
        Secondary allocation = 0 BLOCK
      ]]>
    </ISPF>
  </SERVICE-RESPONSE>
</OPERATIONS-LOG>
<![CDATA[
  Content-type: text/plain

```

```

Entering ISPZINT (Service initialization)
About to read from fileno(stdin) = 0
Data read from STDIN is ISPF TSO DINFO sys1.linklib&SESSION=SPAWN&ISPPROF=USERME.TEST.ISPPROF
EPOCH secs = 1202113932
Local Date & time: Mon Feb  4 03:32:12 2008
Hour: 3 Min: 32 Sec 12
Function ID timestamp = ID012732
Environment variables:
0 QUERY_STRING=
1 CONTENT_TYPE=application/x-www-form-urlencoded
2 PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products/java142/J
3 AUTH_TYPE=Basic
4 DOCUMENT_URI=/ISPZXML
5 SHELL=/bin/sh
6 HTTPS=OFF
7 HTTP_ACCEPT=text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
8 HTTP_USER_AGENT=Java1.3.1
9 SERVER_PORT=1726
10 RULE_FILE=//DD:CONF
11 GATEWAY_INTERFACE=CGI/1.1
12 PATH_INFO=
13 CONTENT_LENGTH=313
14 _CEE_RUNOPTS=ENVAR("_CEE_ENVFILE=//DD:ENV")
15 _BPX_SPAWN_SCRIPT=YES
16 REFERER_URL=
17 _=./ISPZINT
18 CLASSPATH=./usr/lpp/internet/server_root/CAServlet
19 STEPLIB=CURRENT
20 REQUEST_METHOD=POST
21 REMOTE_ADDR=9.190.236.239
22 LANG=C
23 LIBPATH=/bin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin
24 REMOTE_USER=USERME
25 SERVER_ADDR=192.168.123.11

```

```

26 FSCP=IBM-1047
27 PATH_TRANSLATED=
28 HTTP_CONNECTION=keep-alive
29 SERVER_TOKEN=1
30 HTTP_HOST=PTHISD1.AU.IBM.COM
31 _BPX_SHAREAS=YES
32 CGI_ISPCONF=/etc/ispf
33 SERVER_SOFTWARE=IBM HTTP Server/V5R3M0
34 REPORTBITS=77
35 DOCUMENT_ROOT=usr/lpp/ispf/bin
36 NETCP=ISO8859-1
37 CGI_ISPWORK=/var/ispf
38 COUNTERDIR=NULL
39 LC_ALL=en_US.IBM-1047
40 CGI_DTCONF=/etc/ispf
41 _BPX_USERID=USERNME
42 SERVER_PROTOCOL=HTTP/1.1
43 JAVA_HOME=/usr/lpp/products/java142/J1.4/
44 HTTPS_KEYSIZE=
45 TZ=EST5EDT
46 _CEE_ENVFILE=//DD:ENV
47 _BPX_BATCH_SPAWN=SPAWN
48 SCRIPT_NAME=/ISPZXML
49 NLS_PATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/internet/%L/%N:/usr/lib/nls/msg/En_US.IBM-1047/%N
50 CGI_DTWORK=/var/ispf
51 DOCUMENT_NAME=usr/lpp/ispf/bin/ISPZXML
52 SERVER_NAME=PTHISD1.AU.IBM.COM
Number of environment variables is 53
Protocol = HTTP
FSCP = IBM-1047
NETCP = ISO8859-1
CGI_ISPCONF = /etc/ispf
CGI_ISPWORK = /var/ispf
CGI_TRANTABLE =
Server PATH = /bin.:/usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products/java
ISPF standalone function invoked
&ISPPROF value = USERNME.TEST.ISPPROF
ISPF COMMAND = ISPF TSO DINFO sys1.linklib
ISPF PROFILE = USERNME.TEST.ISPPROF
Re-usable ISPF session = SPAWN
About to spawn task for ISPZTSO
Parameters passed to ISPZTSO - PROFILE
Return code from ISPZTSO is 0
About to process PROFILE data in /tmp/USERNME.ID012732.SYSTSPRT
About to malloc() 252 bytes for profdat
*** PROFILE data: 1IKJ56688I CHAR(0) LINE(0) PROMPT INTERCOM NOPAUSE MSGID MODE
WTPMSG NORECOVER PREFIX(USERNME) PLANGUAGE(ENU) SLANGUAG E(ENU) VARSTORAGE(LOW) IKJ56689I
DEFAULT LINE/CHARACTER DELETE CHARACTERS IN EFFECT FOR THIS TERMINAL READY END
Temporary data set prefix set to : USERNME
About to call bpxwdyn to allocate VCMTEMP
Allocating data set USERNME.SCLMDT.VCMISPF.ID012732 to the VCMTEMP DD
1024 bytes of ISPF TSO DINFO sys1.linklib written to VCMTEMP
1024 bytes of /etc/SCLMDTIS;/var/ispf written to VCMTEMP
1024 bytes of /bin.:/usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products/java
Parameter to be passed to ISPZTSO CALL *(ISPZCNT) '+ISPF ID012732 USERNME NONE NONE NONE NONE
Entering Spawn Processing: 03:32:12
PID of this process = 999
RESPfile is /var/ispf/WORKAREA/USERNME/ISPFPID.999
Group PID of this process = 50332642
About to issue system command: ps -u USERNME -o pid,pgid,jobname,xasid,comm >/var/ispf/WORKAREA/USI
PID=6 PGID=50332642 JOBNAME=USERNME ASID=7A COMMAND=/bin/ps
PID=50332642 PGID=50332642 JOBNAME=USERNME ASID=90 COMMAND=/usr/lpp/ispf/bin/ISPZXML
PID=67109859 PGID=50332642 JOBNAME=USERNME ASID=8B COMMAND=/bin/sh
PID=16778213 PGID=50332642 JOBNAME=USERNME ASID=8B COMMAND=./ISPZXENV
PID=998 PGID=50332642 JOBNAME=USERNME ASID=80 COMMAND=/bin/sh
PID=999 PGID=50332642 JOBNAME=USERNME ASID=92 COMMAND=./ISPZINT
PID=16778216 PGID=50332642 JOBNAME=USERNME ASID=7A COMMAND=/bin/sh

```



```

No active ISPF session found - new TSO/ISPF session started
About to issue system command: rm /var/ispf/WORKAREA/USERNME/ISPFPID*
mkdir /var/ispf/WORKAREA/USERNME rc = -1
New SIGfile = /var/ispf/WORKAREA/USERNME/ISPFPID.signal
New CMDfile = /var/ispf/WORKAREA/USERNME/ISPFPID.cmd
RUN directory = /usr/lpp/ispf/bin/
About to spawn task for ISPZTSO
Parameters passed to ISPZTSO - CALL *(ISPZCNT) '+ISPF ID012732 USERNME NONE NONE NONE NONE NONE
Return code from ISPZTSO is 0
Waiting on SIGNAL return 03:32:12
Read signal file: Output = COMPLETE
*** OUTPUT FROM ISPF SESSION ***
Entering ISPZCNT (ISPF Initialization)
Parameters +ISPF ID012732 USERNME NONE NONE NONE NONE NONE NONE NONE USERNME.TEST.ISPPROF ISPF T
REC: sysproc=isp.sisplib,bzz.v1r1.sbzzclib,SYS2.CLIST.ISD1.USER,SCLMDW.V710.SFEKPROC,WD4Z.V710.
Allocation successful for SYSPROC
REC: sysexec=sclmdw.v3.test.exec,USERNME.sclmdw.exec,USERNME.exec
Allocation successful for SYSEXEC
REC: isplib=isp.sispmenu
Allocation successful for ISPLIB
REC: isptlib=isp.sisptenu
Allocation successful for ISPTLIB
REC: isplib=isp.sisppenu
Allocation successful for ISPLIB
REC: ispslib=sclmdw.v3.dev.skels,bzz.v1r1.sbzzsenu,isp.sispsenu,isp.sispslib
Allocation successful for ISPSLIB
REC: isptrace=nullfile
Allocation successful for ISPTRACE
REC: ISPF_timeout = 300
IEBCOPY of ISPF profile RC = 00
NOTE: Data set allocations took 0.49 elapsed seconds
*** XML-NOTE *** Reference tagged SERVICE-RESPONSE
Current Process List:
      PID      PGID  JOBNAME  ASID  COMMAND
50331654  50332642  USERNME   7a  /bin/sh
      10      50332642  USERNME   7a  /bin/ps
50332642  50332642  USERNME   90  /usr/lpp/ispf/bin/ISPZXML
67109859  50332642  USERNME   8b  /bin/sh
16778213  50332642  USERNME   8b  ./ISPZXENV
      998      50332642  USERNME   80  /bin/sh
      999      50332642  USERNME   92  ./ISPZINT
50332648  50332642  USERNME4  89  ./ISPZTSO
]]>
</OPERATIONS-LOG>
<ISPF-OUTPUT>

```

Return and reason codes from the gateway

The RETURN-CODE tag provides the value of the return code from the TSO or ISPF service. If there is an error in the gateway, a value of 8 or 12 is returned with the RETURN-CODE tag and REASON-CODE tags of this format are also provided to describe the error:

```

<REASON-CODE ID="ISPZ0007">ISPF OR SCLM SERVICE HAS ENDED
ABNORMALLY</REASON-CODE>

```

Table 4 lists the possible reason codes when there is a return code of 8 from the gateway:

Table 4. Possible reason codes for a return code of 8 from the gateway

Reason Code	Description
ISPZ0001	Error in ALLOCATION/WRITE to <i>dsname</i> .
ISPZ0002	Processing terminates.
ISPZ0003	Following function has failed: MSG: <i>function</i>

Table 4. Possible reason codes for a return code of 8 from the gateway (continued)

Reason Code	Description
ISPZ0009	Error reading the ISPF.conf configuration file.
ISPZ0010	Ensure file exists in directory specified by the environment variable CGI_ISPCONF in the server configuration.
ISPZ0012	Error in ISPF data set allocation: See error message below
ISPZ0013	Error in allocating the following DD and data set names: MSG: DD= <i>ddname</i> MSG: <i>dsname1 dsname2... dsnamen</i>
ISPZ0014	Verify that the ISPF configuration file ISPF.conf on the host is correct.
ISPZ0224	*** Operation Cancelled ***
ISPZ0225	Possible previous request still active. Either cancel host session or retry once processing terminates.

Table 5 lists the possible reason codes when there is a return code of 12 from the gateway:

Table 5. Possible reason codes for a return code of 12 from the gateway

Reason Code	Description
ISPZ0007	ISPF or SCLM service has ended abnormally.
ISPZ0008	Review log for error details.

Using the native API

This section documents the native API used by some IBM products to invoke the gateway. It is recommended, however, that applications use the XML structured API to invoke the gateway.

Service request format

Service call requests consist of a parameter string passed by means of STDIN to the interface module ISPZINT. The first word of the service request string identifies if the request is for a TSO or ISPF service:

```
TSO service-request
```

or

```
ISPF service-request
```

A TSO service request results in the TSO service running in a TSO address space without the overhead of establishing an ISPF environment. Shown here are some examples of TSO service requests:

```
TSO time
```

```
TSO alloc fi(templ) da('hlq.filetemp') shr
```

An ISPF service request can be for a TSO or ISPF service. The service runs in an ISPF environment established in a TSO address space. The caller can request the ISPF environment be "reusable" (that is, remain active for running subsequent service requests) by including &SESSION=SPAWN in the parameter string. Also the caller can request an existing ISPF profile be dynamically allocated by including &ISPPROF= my.profile.dataset in the parameter string. Shown here are some examples of ISPF service requests:

```

ISPF tso profile
ISPF select cmd(%myexec)
ISPF tso listalc&SESSION=SPAWN&ISPPROF=HLQ.ISPPROF
ISPF tso ex 'my.dataset(exec1) '&SESSION=SPAWN

```

To terminate a "reusable" ISPF session, pass the parameter string shown here to ISPZINT:

```
ISPFFUNC=SHUTDOWN
```

or

```
ISPFFUNC=CANCEL
```

Service response format

Service response data is returned in a tagged format as shown in this section. However this is not valid XML format. Tagged data may be incorporated within system log messages and be stripped out at the client end by means of the encapsulating tags <ISPINFO> </ISPINFO>.

```

System logging data
:
:
<ISPINFO>
ISPF COMMAND : ISPF command issued
RC=X
MSG: xxxxxx
<Specific function request tags>
xxxxxx
xxxxxx
</End of specific function request tags>
</ISPINFO>
System logging data
:
:

```

Service request and response examples

Shown here is the request parameter string and response data returned for an ISPF service request:

Request

```
ISPF TSO PROFILE&SESSION=SPAWN&ISPPROF=&SYSUID..TEST.ISPPROF
```

Response

```

START Transfer DATE/TIME is :Tue Mar 18 12:25:59 WST 2008
At url openConnection
SCLMFunc returned from post is Connect
About to accept response from Server
Response from Server received
Entering ISPZINT (Service initialization)
About to read from fileno(stdin) = 0
Data read from STDIN is ISPF TSO PROFILE&SESSION=SPAWN&ISPPROF=&SYSUID..TEST.ISPPROF
EPOCH secs = 1205814289
Local Date & time: Tue Mar 18 00:24:49 2008
Hour: 0 Min: 24 Sec 49
Function ID timestamp = ID001489
Environment variables:
0 _CEE_ENVFILE=//DD:ENV
1 # PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products/j
2 PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products/jav
3 SHELL=/bin/sh
4 TZ=EST5EDT
5 LANG=C
6 LC_ALL=en_US.IBM-1047
7 NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/internet/%L/%N:/usr/lib/nls/msg/En_US.IBM-1047/
8 LIBPATH=/bin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin
9 # JAVA_HOME=/usr/lpp/products/java531-UK14829/J5.0/bin
10 JAVA_HOME=/usr/lpp/products/java142/J1.4/

```

```

11 CLASSPATH=./usr/lpp/internet/server_root/CAServlet
12 STEPLIB=CURRENT
13 _BPX_BATCH_SPAWN=SPAWN
14 _BPX_SHAREAS=YES
15 # _BPX_SHAREAS=YES
16 # _BPX_SPAWN_SCRIPT=NO
17 CGI_DTCONF=/etc/SCLMDTIS
18 CGI_DTWORK=/var/SCLMDTIS
19 CGI_ISPCONF=/etc/SCLMDTIS
20 CGI_ISPWORK=/var/SCLMDTIS
21 # CGI_TRANTABLE=DOHERTL.LSTRANS.FILE
22 COUNTERDIR=NULL
23 REPORTBITS=77
24 SERVER_SOFTWARE=IBM HTTP Server/V5R3M0
25 SERVER_NAME=PTHISD1.AU.IBM.COM
26 SERVER_PORT=1726
27 _BPX_SPAWN_SCRIPT=YES
28 _BPX_USERID=USERNME
29 RULE_FILE=//DD:CONF
30 SERVER_PROTOCOL=HTTP/1.1
31 REQUEST_METHOD=POST
32 GATEWAY_INTERFACE=CGI/1.1
33 PATH_INFO=
34 PATH_TRANSLATED=
35 QUERY_STRING=
36 SERVER_ADDR=192.168.123.11
37 SERVER_TOKEN=1
38 SCRIPT_NAME=/ISPZINT
39 REMOTE_ADDR=192.168.128.253
40 AUTH_TYPE=Basic
41 REMOTE_USER=USERNME
42 CONTENT_TYPE=application/x-www-form-urlencoded
43 CONTENT_LENGTH=61
44 REFERER_URL=
45 DOCUMENT_ROOT=usr/lpp/ispf/bin
46 DOCUMENT_URI=/ISPZINT
47 DOCUMENT_NAME=usr/lpp/ispf/bin/ISPZINT
48 FSCP=IBM-1047
49 NETCP=ISO8859-1
50 HTTPS_KEYSIZE=
51 HTTPS=OFF
52 HTTP_CONNECTION=keep-alive
53 HTTP_ACCEPT=text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
54 HTTP_HOST=PTHISD1.AU.IBM.COM
55 HTTP_USER_AGENT=Java/1.5.0
56 HTTP_PRAGMA=no-cache
57 HTTP_CACHE_CONTROL=no-cache
58 _CEE_RUNOPTS=ENVAR("_CEE_ENVFILE=//DD:ENV")
Number of environment variables is 59
Connection Protocol = HTTP
Server Name = PTHISD1.AU.IBM.COM
Server Port = 1726
FSCP = IBM-1047
NETCP = ISO8859-1
CGI_ISPCONF = /etc/SCLMDTIS
CGI_ISPWORK = /var/SCLMDTIS
Server PATH = /bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products
ISPF standalone function invoked
&ISPPROF value = &SYSUID..TEST.ISPPROF
ISPF COMMAND = ISPF TSO PROFILE
ISPF PROFILE = USERNME.TEST.ISPPROF
Re-usable ISPF session = SPAWN
About to spawn task for ISPZTSO
Parameters passed to ISPZTSO - PROFILE
Return code from ISPZTSO is 0
About to process PROFILE data in /tmp/USERNME.ID001489.ISPF.SYSTSPRT
About to malloc() 252 bytes for profdat

```

```

Temporary data set prefix set to : USERNME
About to call bpxwdyn to allocate VCMTEMP
Allocating data set USERNME.ISPF.VCMISPF.ID001489 to the VCMTEMP DD
1024 bytes of ISPF TSO PROFILE written to VCMTEMP
1024 bytes of /etc/SCLMDTIS;/var/SCLMDTIS written to VCMTEMP
1024 bytes of /bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products/bin:/usr/lpp/products/sbin:/usr/lpp/products/obj written to VCMTEMP
Parameter to be passed to ISPZTSO CALL *(ISPZCNT) '+ISPF ID001489 USERNME USERNME.TEST.ISPZTSO
Entering Spawn Processing: 00:24:49
PID of this process = 67109485
RESPfile is /var/SCLMDTIS/WORKAREA/USERNME/ISPFPID.67109485
Group PID of this process = 67109485
No ISPZINT.pidlog detected so no existing ISPF session running
No active ISPF session found - new TSO/ISPF session started
About to issue system command: rm /var/SCLMDTIS/WORKAREA/USERNME/ISPFPID*
mkdir /var/SCLMDTIS/WORKAREA/USERNME rc = -1
New SIGfile = /var/SCLMDTIS/WORKAREA/USERNME/ISPFPID.signal
New CMDfile = /var/SCLMDTIS/WORKAREA/USERNME/ISPFPID.cmd
RUN directory = /usr/lpp/ispf/bin/
About to spawn task for ISPZTSO
Parameters passed to ISPZTSO - CALL *(ISPZCNT) '+ISPF ID001489 USERNME USERNME.TEST.ISPZTSO
Return code from ISPZTSO is 0
Waiting on SIGNAL return 00:24:49
Read signal file: Output = COMPLETE
*** OUTPUT FROM ISPF SESSION ***
<ISPINFO>
ISPF COMMAND : SELECT CMD(PROFILE) NEST
RC=0
<ISPF>
IKJ56688I CHAR(0) LINE(0) PROMPT INTERCOM NOPAUSE MSGID MODE WTPMSG NORECOVER
IKJ56689I DEFAULT LINE/CHARACTER DELETE CHARACTERS IN EFFECT FOR THIS TERMINAL
</ISPF>
</ISPINFO>
Current Process List:
      PID      PGID JOBNAME  ASID COMMAND
      94      67109485 USERNME   8a /bin/ps
    50332025   67109485 USERNME3  98 ./ISPZTSO
    16777835   16777835 USERNME   9b ISRUUDL
    67109485   67109485 USERNME   95 /usr/lpp/ispf/bin/ISPZINT
    50332289   67109485 USERNME   8a /bin/sh
RC from doGet then writefile :ok
TOTAL Completion DATE/TIME is :Tue Mar 18 12:26:02 WST 2008

```

Shown here is the request parameter string and response data returned for a TSO service request.

Request

TSO LISTALC

Response

```

START Transfer DATE/TIME is :Tue Mar 18 12:33:55 WST 2008
At url openConnection
SCLMFunc returned from post is Connect
About to accept response from Server
Response from Server received
Entering ISPZINT (Service initialization)
About to read from fileno(stdin) = 0
Data read from STDIN is TSO LISTALC
EPOCH secs = 1205814765
Local Date & time: Tue Mar 18 00:32:45 2008
Hour: 0 Min: 32 Sec 45
Function ID timestamp = ID001965
Environment variables:
0 _CEE_ENVFILE=//DD:ENV
1 # PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products/bin:/usr/lpp/products/sbin:/usr/lpp/products/obj
2 PATH=/bin:./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products/bin:/usr/lpp/products/sbin:/usr/lpp/products/obj
3 SHELL=/bin/sh
4 TZ=EST5EDT

```

```

5 LANG=C
6 LC_ALL=en_US.IBM-1047
7 NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/internet/%L/%N:/usr/lib/nls/msg/En_US.IBM-1047/%N
8 LIBPATH=/bin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin
9 # JAVA_HOME=/usr/lpp/products/java531-UK14829/J5.0/bin
10 JAVA_HOME=/usr/lpp/products/java142/J1.4/
11 CLASSPATH=./usr/lpp/internet/server_root/CAServlet
12 STEPLIB=CURRENT
13 _BPX_BATCH_SPAWN=SPAWN
14 _BPX_SHAREAS=YES
15 # _BPX_SHAREAS=YES
16 # _BPX_SPAWN_SCRIPT=NO
17 CGI_DTCONF=/etc/SCLMDTIS
18 CGI_DTWORK=/var/SCLMDTIS
19 CGI_ISPCONF=/etc/SCLMDTIS
20 CGI_ISPWORK=/var/SCLMDTIS
21 # CGI_TRANTABLE=DOHERTL.LSTRANS.FILE
22 COUNTERDIR=NULL
23 REPORTBITS=77
24 SERVER_SOFTWARE=IBM HTTP Server/V5R3M0
25 SERVER_NAME=PTHISD1.AU.IBM.COM
26 SERVER_PORT=1726
27 _BPX_SPAWN_SCRIPT=YES
28 _BPX_USERID=USERNME
29 RULE_FILE=//DD:CONF
30 SERVER_PROTOCOL=HTTP/1.1
31 REQUEST_METHOD=POST
32 GATEWAY_INTERFACE=CGI/1.1
33 PATH_INFO=
34 PATH_TRANSLATED=
35 QUERY_STRING=
36 SERVER_ADDR=192.168.123.11
37 SERVER_TOKEN=1
38 SCRIPT_NAME=/ISPZINT
39 REMOTE_ADDR=192.168.128.253
40 AUTH_TYPE=Basic
41 REMOTE_USER=USERNME
42 CONTENT_TYPE=application/x-www-form-urlencoded
43 CONTENT_LENGTH=12
44 REFERER_URL=
45 DOCUMENT_ROOT=usr/lpp/ispf/bin
46 DOCUMENT_URI=/ISPZINT
47 DOCUMENT_NAME=usr/lpp/ispf/bin/ISPZINT
48 FSCP=IBM-1047
49 NETCP=ISO8859-1
50 HTTPS_KEYSIZE=
51 HTTPS=OFF
52 HTTP_CONNECTION=keep-alive
53 HTTP_ACCEPT=text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
54 HTTP_HOST=PTHISD1.AU.IBM.COM
55 HTTP_USER_AGENT=Java/1.5.0
56 HTTP_PRAGMA=no-cache
57 HTTP_CACHE_CONTROL=no-cache
58 _CEE_RUNOPTS=ENVAR("_CEE_ENVFILE=//DD:ENV")
Number of environment variables is 59
Connection Protocol = HTTP
Server Name = PTHISD1.AU.IBM.COM
Server Port = 1726
FSCP = IBM-1047
NETCP = ISO8859-1
CGI_ISPCONF = /etc/SCLMDTIS
CGI_ISPWORK = /var/SCLMDTIS
Server PATH = /bin./usr/sbin:/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/products
About to spawn task for ISPZTSO
Parameters passed to ISPZTSO - LISTALC
Return code from ISPZTSO is 0
About to open /tmp/USERNME.ID001965.ISPF.SYSTSPRT

```

```
<ISPINFO>
<TSO>
PDFTDEV.USERNME.LOAD
PDFTDEV.STG.LOAD
PDFTDEV.INT.LOAD
PDFTDEV.SVT.LOAD
USERNME.SCLMDW.LOAD
SCLMDW.V3TEST.LOAD
SCLMDW.V3BASE.LOAD
/tmp/USERNME.ID001965.ISPF.SYSTSPRT
</TSO>
</ISPINFO>
RC from doGet then writefile :ok
TOTAL Completion DATE/TIME is :Tue Mar 18 12:33:56 WST 2008
```




Printed in USA

Spine information:



Interactive System Productivity
Facility (ISPF)

TSO/ISPF client gateway (z/OS V1 R8.0)