The Mainframe Software Partner
For The Next 50 Years

Host Communications Interface

User/Reference Guide
z/OS Version

Release 3.0

Global technical support contact information is available from our Customer Support Web Site:
http://go.compuware.com

Doc. CWHCUX3A Rev. 11
April 8, 2015.

Contents

Figures

Tables

Introduction

The Host Communications Interface User/Reference guide provides reference information for upgrading, customizing, and using Host Communications Interface (HCI).

This guide is intended as a reference to technical information for the HCI. For details on installing and configuring the HCI when supporting a specific Compuware product, refer to that product's documentation for more information.

Manual Structure

The following chapters comprise the *Host Communications Interface User/Reference Guide:*

- **Chapter 1, "HCI Overview":** Provides an overview of the Host Communications Interface (HCI) product.
- **Chapter 2, "Migration Considerations":** Describes the HCI Release 3.0 enhance and the installation migration procedure.
- **Chapter 3, "Executing the HCI":** Illustrates with sample JCL how to execute the HCI as either a batch job or started task.
- **Chapter 4, "Using the Server Activation Facility":** Explains the Server Activation Facility and how to use it.
- **Chapter 5, "Implementing Parallel Sysplex Support":** Describes how to install the HCI with parallel sysplex support.
- **Chapter 6, "Using the Journaling Facility":** Describes the HCI journaling facility and illustrates with sample JCL how to allocate and initialize journal dataset.
- **Chapter 7, "Operator Communications":** Explains the HCI operator commands.
- **Chapter 8, "HCI Diagnosis and Debugging":** Describes how to invoke a dump, prepare journal data, and send them to Compuware Technical Support for help in diagnosing any HCI problems.
- **Chapter 9, "Using the Source Parameter Facility":** Explains the HCI Source Parameter Facility and how to use it to create XML documents for specifying HCI parameters.
- **Chapter 10, "Using HCI-XPERT":** Describes the HCI-XPERT diagnostic tool and its utilities and how to use them.
- **Appendix A, "Sample XML Parameters":** Provides a sample XML document of HCI elements and their attributes.

Intended Audience

This manual is intended for mainframe product installers and network management personnel.

Related Publications

Refer to the Enterprise Common Components documentation set that is included on the product media for more information about the Compuware Shared Services, Licensed Management System, and Host Communications Interface products. See below for access to additional product documentation.

Online Documentation

The product installation media includes the following Host Communications Interface documentation in electronic format:

- Release Notes in HTML format
- Product manuals in PDF format
- Adobe PDF index file (PDX file).

The product manuals are also available in the following electronic formats on Compuware's FrontLine customer support Web site for viewing or downloading:

- View PDF files with the free Adobe Reader, available at **http://www.adobe.com**.
- View HTML files with any standard Web browser.

Getting Help

Compuware provides a variety of support resources to make it easy for you to find the information you need.

Compuware Go Customer Support Website

You can access online information for Compuware products via our Compuware Go customer support website at **http://go.compuware.com**.

Compuware Go provides access to critical information about your Compuware products. You can review frequently asked questions, read or download documentation, access product fixes, or e-mail your questions or comments. The first time you access Compuware Go, you are required to register and obtain a password. Registration is free.

Compuware now offers User Communities, online forums to collaborate, network, and exchange best practices with other Compuware solution users worldwide. To join, go to **http://groups.compuware.com**.

Contacting Customer Support

Phone

- USA and Canada: 1-800-538-7822 or 1-313-227-5444.
- All other countries: Contact your local Compuware office. Contact information is available at **http://go.compuware.com**.

Web

You can report issues via the Quick Link **Create & View Support Cases** on the Compuware Go home page.

Note: Please report all high-priority issues by telephone.

Mail

Compuware Customer Support
Compuware Corporation
One Campus Martius
Detroit, MI 48226-5099

Corporate Website

To access Compuware's site on the Web, go to **http://www.compuware.com**.

The Compuware site provides a variety of product and support information.

Customer Support

Compuware continually strives to improve our software products and documentation. Feedback from our customers helps us to maintain our quality standards.

While the software should execute according to documented specifications, there are times when problems do occur. If HCI problems arise, see Chapter 8, "HCI Diagnosis and Debugging" for information about how to generate HCI data that will assist the Customer Support personnel in determining the exact cause of your problem in a timely manner.

Compuware Customer Support
http://frontline.compuware.com

Chapter 1.
HCI Overview

What's New in This Release

Enhancements for Revision 9 of this document include discussions of new support for the z/OS Automatic Restart Management Service (ARM). This support in the operating system can be found in IBM manual *z/OS VnnRnn MVS Sysplex Services Reference,* document number SA22-7618. At the time of this publication, z/OS V1R13.0 and version 14 of the manual were used as references. Prior to adding the new parameter(s) to the HCI in order to invoke this ARM facility, Compuware recommends reading through the procedures this manual, and any manuals referenced, to gain a better understanding of what is required to implement ARM support into the operating system.

Note: In order for ARM processing to record every execution of the HCI, you must add the following operands to the //HCIPRINT DD statement:

FREE=CLOSE
SPIN=UNALLOC

(Do NOT add them to the //HCIERR DD statement)

When implementing HCI in support of the Compuware Workbench, a TSO/ISPF facility, the Host Communication Configuration Facility (HCCF), is provided to automate the generation of the HCI configuration parameters, datasets, and execution JCL. Refer to the *Compuware Workbench Installation Guide* for more information.

Enhancements for Revision 7 and 8 of this document were editorial improvements for readability. Note that the specification of the //HCIERR DD statement must specify SYSOUT=* as this is an output DD. The section documenting the IPv6 support in the HCI has been enhanced to provide a more complete discussion of its use.

Enhancements in Revision 6 of this document changed the instructions for defining the HCIXML DD statement and the discussion of the XML Dynamic Refresh capability. The discussion has been augmented based upon feedback from the development organization. The new DD definition matches the implementation in the HCI and removes the requirement that any changes be made to the HCI JCL simply due to applying the PTF. Only if the REFRESH XML=xxxxxxxx command is used must the HCIXML DD be defined as specified in this revision to this document.

Enhancements in Revision 5 of this document includes a discussion of using XML to provide a dynamic refresh capability. This feature has been present only for the macro parameter facility, but this facility is extended to include the XML parameters.

Major enhancements in Host Communications Interface (HCI) 3.0 include:

- Support for TCP/IP IPv6 communication
- Support for TCP/IP DVIPA communication
- Addition of source-form parameters supplied at HCI startup via an XML document
- Support for z/OS system variable substitution in startup parameters.

Minor HCI enhancements have been made to enhance ongoing maintenance and to provide better diagnostics.

Refer to Chapter 2, "Migration Considerations" for the Compuware-recommended procedure for upgrading from HCI Release 2.5 to 3.0.

Product Overview

Compuware's Host Communications Interface (HCI) is a communications middleware product that provides a common API to Compuware mainframe products for TCP/IP (IPv4 and IPv6), LU 6.2, LU 2, and LU 0 communications with network or SYSPLEX connected partner programs.

One HCI region on each LPAR can support any number of Compuware products, or there may be a dedicated HCI instance for each installed product. For example, DevEnterprise might have its own HCI running, while the Compuware Workbench might have its own. With the Source Parameter Facility, combining multiple HCI regions into a single region becomes more manageable.

SYSPLEX support exists whereby a single HCI region on one member of the SYSPLEX can accept and transfer connection requests to another member of the SYSPLEX. Communication with the network always occurs from the HCI instance accepting the original request.

HCI can act as a *listener* task waiting for connection requests from network-connected products. It can submit JCL or issue a START command to initiate a program designed to handle the connection request and subsequent data transmissions. Alternatively, the HCI can ATTACH/ATTACHX Compuware products such that they do not consume an address space, but instead execute within the HCI address space itself. Further, the HCI can act as the communications pipeline for HTML data going to and coming from client browsers into application server programs.

Data security is provided for TCP/IP using the TLS facility of the protocol stack. HCI provides no encryption or decryption of application data, but does allow the protocol stack to perform these functions. Thus, all encryption techniques that are supported by TCP/IP can be employed without changing any application program code.

In addition to listening for connection requests, the HCI can initiate outbound requests under control of application programs using only the single API. As such, HCI can act as both a server and a client for multiple connections/conversations using multiple communication protocols. Additional features can include APIs for establishing environments running under particular user IDs, in order to protect data from unauthorized access.

A logging (journal) facility exists into which a TP application program can write any information it chooses. The HCI uses this log to provide diagnostic information useful to Compuware developers. If a problem occurs that cannot be diagnosed by other means, Compuware may request that you enable the log/journal and capture information from the time of the problem.

Error and informational messages are internationalized and can be localized by translating the messages in a single source data set (not allowed by customers). Error messages generated by the XML parsing facility are not currently internationalized in this manner.

Operating System Requirements

HCI 3.0 requires z/OS Release 1.8 or more current. Future enhancements to the HCI may require later releases of the operating system.

IPv6 Support

If IPv6 communication is required, it must be enabled in the z/OS TCP/IP region(s) that support for the HCI. Enabling IPv6 in the protocol stacks does not preclude IPv4

connections. HCI supports both IPv4 and IPv6 simultaneously, on a connection-by-connection basis. Hence, a migration to IPv6 can occur without requiring changes to HCI code.

IPv6 is disabled by default, but can be enabled by specifying the **IPV6="YES"** attribute(s) in one or more HCICNAMB elements. IPv6 is not supported if a load module configuration is used that has not been reassembled using the HCI 3.0 macro libraries, but can be enabled if the macro form parameters are reassembled.

A new parameter has been added to the HCICNAMB element. You must specify **IPV6="YES"** in each HCICNAMB element for which you want IPv6 support. You may have a mixture of **IPV6="YES"** and **IPV6="NO"** HCICNAMB elements. That is, there is no need for IPv6 to be enabled for every TCP/IP region used by the HCI. If your installation has multiple z/OS TCP/IP regions, and only one is enabled for IPv6, then it would make sense to specify **IPV6="YES"** only for this region. In this way, you can test the HCI's IPv6 support while still using the IPv4 TCP/IP region.

DVIPA Support

VIPARANGE support is enabled by the addition of the DVIPA or DVIPN attributes in the HCICNPCB elements and must be configured in the TCP/IP region parameters. Refer to the appropriate IBM manual in the Communications Server bookshelf for information about this feature of TCP/IP.

Source Parameter Facility

The Source Parameter Facility is implemented by the z/OS XML system parser, which must be available to the HCI. Do not confuse the system parser with the XML toolkit (FMID HXML180 as of this publication), which is a separate addition to z/OS. The toolkit is not required for HCI execution.

The Source Parameter Facility allows the input of HCI execution parameters directly into the HCI without requiring a previous assembly and link job. These parameters are specified as an XML document, which requires the z/OS system parser.

The Source Parameter Facility is optional, and the assembled and linked configuration module is still supported by HCI 3.0. Conversion to the XML parameters is at your site's discretion.

System Variable Substitution Support

Before invoking the z/OS system parser, each XML element value is processed by the z/OS System Variable Substitution Facility. Hence any variables defined either by z/OS or by a particular installation can be invoked in the source parameter. For instance, the system variable &SYSNAME is always initialized by z/OS to be the same name the installation designated for the particular operating system image (LPAR). By including the character string `&SYSNAME` in an XML source statement (for example, in the name of the log dataset, when the HCI is started) the name of the particular LPAR replaces the `&SYSNAME` string in the XML, which is subsequently parsed. Hence, the installation site does not need to keep a separate XML source document for each LPAR that runs the HCI. Instead, each HCI has its own log dataset name.

Chapter 2.
Migration Considerations

Note: Migration procedures described in this chapter apply to upgrading to HCI 3.0 from Release 2.5. Whereas most steps also apply when upgrading from Release 2.4, Compuware cautions that unpredictable results may occur.

Upgrading your site to HCI 3.0 can be accomplished by changing the dataset(s) named in the //STEPLIB or //JOBLIB DD used for HCI execution. If none of the new HCI features are required, then no changes need be made to existing parameters or to execution JCL. However, Compuware recommends that the numbered steps in "Migration Procedure" on page 2-2 be followed over some period of time so that the full set of new HCI features can be implemented.

New DD Statements

The //HCIPRINT DD is optional. The //HCIERR and //HCIXML DD statements are required only if you use the new Source Parameter Facility.

You can add the following new DD statements to the HCI execution JCL:

//HCIPRINT DD SYSOUT=*, FREE=CLOSE, SPIN=UNALLOC	Generates and prints an optional, new complete diagnostic report containing the HCI parameter listing, load module information, and system environmental information. DCB characteristics for this dataset are: DCB=(RECFM=FBA,LRECL=153). Messages are written only to this dataset, not to the console or job log. The FREE= and SPIN= operands are required if you are using ARM support to view all of the HCIPRINT data sets (one for each of the automatically restarted instances).
//HCIERR DD SYSOUT=*	Displays detailed error messages concerning any XML parameter errors or inconsistencies only. DCB characteristics for this dataset are: DCB=(RECFM=FB,LRECL=80). The HCIERR data set contains *only* errors detected in the XML document and *not* errors detected in the pre-assembled/linked configuration load module. Do not add the FREE or SPIN operands to the //HCIERR DD statement.
//HCIXML DD * ***or*** **//HCIXML DD DSN=**hlq.HCI.XML(*member*)**, DISP=SHR**	Supplies the XML document parameters via the new source format parameter facility. Overrides the use of the preassembled/linked configuration load module and uses the XML document instead. DCB characteristics for this dataset are: DCB=(RECFM=FB,LRECL=80). Include the parameters directly into the JCL or specify a sequential dataset or a member of a partitioned dataset.

Source Parameter Facility

Migrate from the macro format, preassembled and link-edited configuration parameters to the new Source Parameter Facility. Compuware recommends that you convert at your earliest convenience.

The Source Parameter Facility is enabled by the addition of the //HCIXML DD statement in the HCI execution JCL. When this DD statement is detected, the load module parameters are ignored and the XML parameters from this dataset are used instead. Removing this DD statement causes the HCI to revert to the old load module parameter (FDBDRPL DD).

Creating the Source Parameters

The source parameters are supplied via the //HCIXML DD statement and are parsed by the HCI using the z/OS system XML parser. Therefore, they must constitute a well-formed XML document. Because manually creating this document would be time consuming and prone to error, the HCI provides a mechanism to automatically create the XML statements, which correspond to a given assembled configuration load module.

Migration Procedure

Follow these steps to create the XML source. You need to perform this procedure only once. Once you have created the XML dataset, it can be edited and used from then on. You can manually alter this XML. The procedure below simply constitutes a starting point for creating the XML.

1. Add a //HCIPRINT DD SYSOUT=* statement to your HCI execution JCL.
2. Execute the HCI.
3. Terminate the HCI and examine the HCIPRINT dataset.
4. Copy and paste the XML, starting with the <HCICNFIG> statement through the </HCICNFIG> statement into a member (HCIMCNFG) of an 80-byte source partitioned dataset. Ensure that you copy all of the XML.
5. Add an //HCIXML DD statement pointing to the member of the 80-byte partitioned dataset you just created.
6. Add an //HCIERR DD SYSOUT=* statement to the HCI execution JCL.
7. Execute the HCI.

Notice that the HCIERR dataset contains a listing of all the XML statements with error messages describing any invalid entries. At the end of this dataset, if no errors have been detected, the following message is printed:

```
HC1860 NO STATEMENTS FLAGGED IN THE XML DOCUMENT
```

System Variable Substitution

Before the HCI processes any attribute value parsed from the XML document, the value is passed to the z/OS System Parameter Substitution facility. If you have enabled this facility in the operating system, insert a valid substitution variable (&***) into the value of an XML element. If the substitution variable is defined to your system, its value is replaced in the value of the attribute element.

For example, if you want to name your HCI journal datasets with the SYSNAME of the LPAR on which the HCI is running, specify the DSNAME attribute of an HCICNJCB element like this:

```
DSNAME="CPWR.HCI.&SYSNAME..JOURNAL1"
```

&SYSNAME is replaced with the one- to eight-character name of your z/OS LPAR. Note the double period following the variable &SYSNAME. Also note that the resulting attribute value (after substitution) must fit in the same 72-byte XML source statement. You cannot automatically continue an XML statement by using parameter substitution.

Chapter 3.
Executing the HCI

The HCI can be submitted as a batch job or a started task. See the sample JCL below.

Batch Job Execution

Tailor the HCI batch job execution JCL example, shown in Figure 3-1, for your installation.

Figure 3-1. Batch Job Execution JCL

```
//*
//*  Place your JOB statement here
//*
//HCISTEP  EXEC PGM=HCIMMAIN,REGION=0M,TIME=NOLIMIT,
//              PARM="SYSID=xxxx,PORT=nnnnn,CONFIG=xxxxxxxx"
//STEPLIB   DD DSN=hlq.HCI.AUTHLIB,DISP=SHR
//HCIPRINT  DD SYSOUT=*
//HCIPARM   DD DSN=hlq.HCI.PARMLIB,DISP=SHR
//HCIJCL    DD SYSOUT=(A,INTRDR)
//HCIERR    DD SYSOUT=*
//FDBDRPL   DD DSN=hlq.HCI.CONFIG.LOAD,DISP=SHR
//HCIXML    DD DSM=hlq.HCI.XMLLIB(member),DISP=SHR
```

Started Task Execution

Store the started task JCL in SYS1.PROCLIB or an installation appropriate procedure library. Tailor the HCI started task execution JCL example, shown in Figure 3-2, for your site.

Figure 3-2. Started Task Execution JCL

```
//HCIPROC  PROC
//HCISTEP  EXEC PGM=HCIMMAIN,REGION=0M,TIME=NOLIMIT,
//              PARM="SYSID=xxxx,PORT=nnnnn,CONFIG=xxxxxxxx"
//STEPLIB   DD DSN=hlq.HCI.AUTHLIB,DISP=SHR
//HCIPRINT  DD SYSOUT=*
//HCIPARM   DD DSN=hlq.HCI.PARMLIB,DISP=SHR
//HCIJCL    DD SYSOUT=(A,INTRDR)
//HCIERR    DD SYSOUT=*
//FDBDRPL   DD DSN=hlq.HCI.CONFIG.LOAD,DISP=SHR
//HCIXML   DD DSN=hlq.HCI.XML(member),DISP=SHR
```

JCL Notes

This section describes the HCI execution JCL.

JOB Statement

Your installation's requirements for the JOB statement dictate how it must look. Ensure that you specify REGION=0M on the JOB statement as well as later on the EXEC statement. Because the HCI is a long-running address space, either specify a job CLASS that will not time out, or specify TIME=NOLIMIT.

EXEC Statement

In order for the HCI to manage its environment efficiently, code REGION=0M and TIME=NOLIMIT on the EXEC statement. The HCICNGCA element attributes give you the opportunity to limit the amount of virtual storage consumed by the HCI so there is no reason to limit the storage with the REGION parameter as well. The HCI abends if it issues a request for virtual storage and that request is not satisfied due to the REGION limit. Coding TIME=NOLIMIT allows the HCI to remain executing for as long as you desire.

The PARM= operand on the EXEC statement is optional. If included it must be coded as shown with the three sub-operands enclosed in double quotes:

- **SYSID=xxxx:** Specify the four-character subsystem ID of the HCI. This parm overrides the SYSID=xxxx attribute in the XML parameter's <HCICNGCA element or the SYSID=xxxx operand on the HCICNGCA macro.
- **PORT=nnnnn**: Ignored in HCI 3.0.
- **CONFIG=xxxxxxxx**: If you are using the macro-generated load module configuration facility from HCI 2.5, you can specify the name of the load module in the //FDBDRPL dataset that is to be used for this execution of the HCI. The default is **HCICNFIG** if this suboperand is omitted. If the //HCIXML DD is present, this suboperand is ignored.

//STEPLIB DD

Compuware recommends that you place the HCI load modules in a dataset named in the //STEPLIB (or //JOBLIB) DD statement. Although you may install the HCI into a LINKLIST dataset, Compuware does not recommend that you do so. By using a //STEPLIB DD statement, you isolate your site’s installation from problems caused by future releases of the HCI. All datasets in the HCI //STEPLIB dataset must be APF-authorized.

//HCIPRINT DD

The //HCIPRINT DD statement is an output dataset that can name a SYSOUT class which contains diagnostic information previously written to the system console. Additional information concerning the HCI, including an XML representation of your configuration, can be found in this dataset as well. If you contact Compuware technical support for any reason, have the contents of this dataset available.

//HCIPARM DD

The //HCIPARM DD statement is an input partitioned dataset from which the HCI obtains JCL used by the Server Activation Facility. Each member of this PDS must be named in a MEMBER= attribute of one or more of the HCICNTPT elements.

//HCIJCL DD

The //HCIJCL DD statement is an output dataset which must be coded exactly as shown. This is the dataset to which the HCI writes JCL statements to be submitted via the Server Activation Facility.

//HCIERR DD

The //HCIERR DD statement is an output dataset into which the HCI copies the XML document that was used for configuration and into which all error messages concerning the format or content of the XML are written. If the //HCIXML DD statement is present, then the //HCIERR DD statement is required. Otherwise, this DD is ignored.

//FDBDRPL DD

During migration, when the XML dataset has not yet been created, the //FDBDRPL DD is an input data set from which the configuration load module is obtained (the //FDBDRPL DD was previously created by the macro assembly and link edit). The name of this load module can be specified in the CONFIG=xxxxxxxx suboperand of the EXEC statement. If the //HCIXML DD is present, this DD is ignored.

//HCIXML DD

Supply the XML document parameters (via the new source format parameter facility) by adding the following DD statement to the HCI execution JCL:

```
//HCIXML    DD DSN=hlq.HCI.XML(member),DISP=SHR
```

The presence of this DD statement will override the use of the pre-assembled/linked configuration load module (//FDBDRPL DD) and will cause the XML document to be used instead. DCB characteristics for this data set are as follows: DCB=(RECFM=FB,LRECL=80). This data set can be sequential, containing just the initial XML document which configures the HCI. Or the source parameter data set can be inline data via a //HCIXML DD * statement. In either of these cases, the REFRESH XML=xxxxxxxx command is NOT available to you, but these forms of the DD statement are allowed for migration purposes for your initial HCI 3.0 implementation.

The //HCIXML DD is not required for HCI 3.0, which allows you to update to HCI 3.0 without changing your JCL until you are ready to migrate to the new XML parameters.

However, if you want to use the REFRESH XML=xxxxxxxx operator command, this data set must be a partitioned data set and the member containing the initial parameters must be specified as the member name on the DSN = parameter. The REFRESH command names the member of this data set which is to be used in the refresh operation. See example #3 below

Examples:

Example #1: XML is inline and REFRESH XML=xxxxxxx is NOT supported:

```
//HCIXML    DD  *
<HCICNFIG>
 ……xml statements
</HCICNFIG>
/*
```

Example #2: XML is in a sequential data set and REFRESH XML=xxxxxxxx is NOT supported:

```
//HCIXML    DD   DSN=hlq.XML.DATA,DISP=SHR
```

Example #3: XML is in a partitioned data set and the REFRESH XML=xxxxxxxx IS supported:

```
//HCIXML     DD   DSN=hlq.HCI.XML(HCIMCNFG),DISP=SHR
```

Additional members can be added to this PDS and can be specified in a subsequent REFRESH XML=xxxxxxxx command. In this example, the member named HCIMCNFG will be used for initial configuration at HCI startup. Other members of this PDS can be activated via the REFRESH XML=xxxxxxxx command.

The data set names and member names shown above are only examples. You may choose any valid names that you wish.

Starting the HCI Automatically at IPL Time

You may place the PROC that starts the HCI into the list of procedures, which are automatically started by z/OS at IPL time. However, ensure that VTAM, TCP/IP, VSAM, RACF and the Compuware License Management System are all initialized before starting the HCI.

Chapter 4.
Using the Server Activation Facility

The Server Activation Facility (SAF) is a portion of the HCI that is concerned with allocating new conversations and/or connections to existing server applications and starting new server applications when there is not already a server application that can accept another connection or conversation.

Characteristics of the Server Activation Facility:

A single server program can support any number of conversations/connections simultaneously.

A *generic* TP_NAME facility allows multiple servers, all with different TP_NAMES, to be known by a single generic name to LUs on the network (LU 6.2 only).

An *alias* TP_NAME facility allows servers with the same TP_NAME to be known by multiple alias names.

A server program can start and register itself with the HCI and, thereafter, receive all conversations bound to its TP_NAME. A server program is bound to a z/OS TCB, instead of to a z/OS ASCB, thus allowing the HCI to clean up resources held by the server when the TCB terminates, rather than having to wait for the address space to terminate.

In addition to submitting a batch job or to issuing the z/OS START command, the HCI can attach a TP program within the HCI address space. This facility is designed for TP programs that need to start quickly and that can manage their own system resources without requiring any JCL allocation facilities.

Controlling the Server Activation Facility

Parameters to the Server Activation Facility exist as members in the HCI PARMLIB dataset and as entries in the configuration assembly member. Static parameters define each TP in the HCICNTPT macro or XML element as to its execution characteristics. See "<HCICNTPT Element" on page 9-21 for more information.

TP_PROFILE_TABLE (TPT) — Parameters in the HCICNTPT Configuration

The TP_PROFILE_TABLE provides the following information:

- An indicator as to whether this is a generic TP_NAME
- An indicator as to whether this is an alias TP_NAME
- PARMLIB member name containing initiation instructions
- Maximum number of conversations per instance of TP_NAME
- Maximum number of instances of TP_NAME
- Maximum time job queued but not yet submitted
- Maximum time job submitted but not yet started.

From this information, the HCI can submit instances of this server, allocate as many conversations as desired to the existing instance, and can then submit another instance.

If the TP_NAME is generic, it signifies that there will be multiple instances of the server, each prestarted (as opposed to using the submit facility), and that each one of them has same TP_NAME for the length specified, and has unique values for the rest of the TP_NAME. As conversation allocation requests arrive, they will contain the same generic TP_NAME. The Server Activation Facility allocates the conversations to the existing applications limited by the two maximum values contained in the TPT.

Note that an entry in the HCI PARMLIB dataset is not required for servers that do not require the generic facility capability nor need to be started automatically. The necessary control blocks are created when the application TP registers

Information needed by the HCI to actually initiate a TP is contained within the member of the PARMLIB dataset named by the MEMBER= attribute of the configuration element.

Batch Jobs

The HCI submits JCL to the JES2 or JES3 internal reader in order to initiate TP programs as batch jobs. JCL within the PARMLIB member is identified by the presence of forward slashes (//) in the first two positions of the first statement in the member. The first statement is the z/OS JOB. The job name is overridden and a unique name is substituted. This unique name is constructed using the job name prefix specified in the HCICNGCA configuration element. A number is suffixed to this prefix to make this name unique and eight characters in length.

Sysplex Routed Tasks

Routed task support is part of the sysplex implementation and is always active when the SYSPLEX attribute has been specified through HCICNGCA. The HCI issues the z/OS ROUTE command in order to initiate server applications. To use the routed task support, the HCI PARMLIB member named in the HCICNTPT entry must contain at least a single record. The word *ROUTE* must be in column one, followed by either

> One or more blanks, and then the name of the procedure, which must exist in the system PROCLIB

or

> A list of sysplex z/OS names delimited by commas and enclosed in parentheses, followed by one or more spaces, and then the name of the procedure, which must exist in the system PROCLIB.

Other records are ignored for either of the above.

If no list is specified, the ROUTCMD= configuration parameter is checked. If **ROUTCMD=NO** was specified, the ROUTE is changed to a simple START. This change results in the task being started on the same z/OS on which the HCI is running. If **ROUTCMD=YES** was specified, all active z/OS systems are evaluated, and the one currently least routed to for HCI purposes is selected.

If HCI sysplex support is not activated, all ROUTE statements are converted to a simple START. Therefore, any route list is omitted.

ROUTE(System_List)

The route system list is evaluated as to the number of tasks that have been routed to each system. The one with the least number is chosen. The sysplex z/OS names can be

specified as a prefix name; that is, any z/OS name having the same prefix becomes part of the list. A prefix name is specified by suffixing an asterisk (*) to the prefix. For example, **CW*** indicates any z/OS name starting with *CW* will be added to the list. This format allows the specification of a group of z/OS systems with one single list entry.

Short notation of a single asterisk (*) can be used to denote all z/OS systems in the sysplex. This format is used alone with no other names. In addition, you can group entries in the list by using a semicolon (;) instead of a comma (,).

The groups are processed from left to right. If all z/OS systems in a group are inactive, the next group is evaluated. Again, within a group of one or more active z/OS systems, the least routed is selected. Using this grouping technique, you can establish a priority list. For example, this grouping evaluates from left to right and selects the first active z/OS system. :

```
(CW03;CW05;CW01)
```

Whereas, this list selects the one with the least number of routes to it:

```
(CW03,CW05,CW01)
```

You can mix prefix names and grouping in the list. If the last is a group, it is not necessary to use a semicolon (;) prior to the ending parenthesis or comma (,) if it is not a group.

The system list is denoted by the presence of the left and right parenthesis. This list must follow the ROUTE without spaces. Spaces within the list will be compressed out and are not used as if a comma.

If none of the z/OS systems named are active, it is treated as an error. If the sysplex has been changed to a monosystem, the ROUTE is changed to a START if the system list contained the z/OS system ID of the system that the HCI is currently running on. The HCI performs the following:

1. A unique name is constructed using the job name prefix specified in the HCICNGCA configuration macro, and to this prefix is suffixed a number to make this name unique. The unique name is used as the value for the START keyword JOBNAME=. This change is done to support those sites using Master JCL. The unique name is placed after the PROC named in the PARMLIB member.
2. Following the unique name, the HCI places the following:

   ```
   ,HCISYS=xxxx
   ```

   where xxxx is replaced with the SUBSYSTEM_ID.
3. Following this, the HCI places the following:

   ```
   ,HCIUSER=xxxxxxxx
   ```

   where xxxxxxxx is replaced with the user ID of the client that initiated the conversation request.
4. Following this, the HCI places the following:

   ```
   ,HCIGRP=xxxxxxxx
   ```

   where xxxxxxxx is replaced with the group name of the client that initiated the conversation request.

   The z/OS MCGR macro is issued with the above command.

Note that items 3 and 4 above, apply only to LU 6.2 conversations, and not to TCP/IP connections.

For example, assume an HCI PARMLIB member named SYPERVR contains the following:

```
ROUTE(CW01) SRVRPROC
```

The HCICNGCA macro specifies a SUBSYSTEM_ID of HCIA, and a JOBNAME prefix of HCIJOB, and ROUTCMD=YES. A PROC named SRVRPROC exists in SYS1.PROCLIB.

The HCI issues the following command:

```
ROUTE CW01,START SRVRPROC,JOBNAME=HCIJOB01,HCISYS=HCIA,
HCIUSER=*userid,HCIGRP=*group
```

The PROC has the following requirements:

- The first step in this PROC must execute a program named HCISERVR.
- This program must be passed the parameter string, which is the HCI SUBSYSTEM_ID, followed by the user ID, followed by the group name.
- Subsequent step(s) in the PROC invoke the user's TP program.

Sample JCL

```
//IEFPROC PROC
//HCISTEP EXEC PGM=HCISERVR,PARM='&HCISYS,&HCIUSER,&HCIGRP'
//STEPLIB DD DSN=users.authorized.hci.loadlib,DISP=SHR
//USRSTEP EXEC PGM=userpgm
//USRDDS  DD statements as required by userpgm
```

The program, HCISERVR, establishes the security environment appropriately for the execution of the user's server program. That is, the user's program executes under the user ID of the originating client user at the workstation. This environment matches the one that would exist if the user's program were submitted as a batch job instead of being started as a started task.

If HCISERVR encounters any problems, it abends and, thus, the PROC terminates. The HCI detects that the server has not registered within the allotted time and notifies the client LU that the conversation cannot be started.

HCISERVR abends with a U0100 abend and with a reason code identifying the reason for the abend.

Started Tasks

The HCI issues the z/OS START command in order to initiate server applications. To use the started task support, the HCI PARMLIB member named in the HCICNTPT entry must contain at least a single record. The word *START* must be in column one, followed by one or more blanks, and then the name of the procedure, which must exist in one of the system PROCLIBs. Other records are ignored. The HCI performs the following:

1. A unique name is constructed using the job name prefix specified in the HCICNGCA element, and to this prefix is suffixed a number to make this name unique. The unique name is used as the value for the START keyword JOBNAME=. This is done to support those sites using Master JCL. The unique name follows the PROC named in the PARMLIB member.
2. Following the unique name, the HCI places the following:

   ```
   ,HCISYS=xxxx
   ```

   where xxxx is replaced with the SUBSYSTEM_ID.

   The z/OS MCGR macro is issued with the above command.

For example, assume an HCI PARMLIB member named STSERVR contains the following:

```
START SRVRPROC
```

The HCICNGCA macro specifies a SUBSYSTEM_ID of HCIA and a JOBNAME prefix of HCIJOB. A PROC named SRVRPROC exists in SYS1.PROCLIB.

The HCI issues the following command:

```
START SRVRPROC.HCIJOB01,HCISYS=HCIA
```

The PROC, which the user must have placed in the system PROCLIB, has the following requirements:

- The first step in this PROC must execute a program named HCISERVR.
- This program must be passed a single parameter, which is the HCI SUBSYSTEM_ID.
- Subsequent step(s) in the PROC execute the user's server program.

Sample JCL

```
//IEFPROC PROC
//HCISTEP EXEC PGM=HCISERVR,PARM=&HCISYS
//STEPLIB DD DSN=users.authorized.hci.loadlib,DISP=SHR
//USRSTEP EXEC PGM=userpgm
//USRDDS  DD statements as required by userpgm
```

For LU 6.2 conversations only, the program, HCISERVR, establishes the security environment appropriately for the execution of the user's server program. That is, the user's program executes under the user ID of the originating client user at the workstation. This environment matches the one that would exist if the user's program were submitted as a batch job instead of being started as a started task.

If HCISERVR encounters any problems, it abends and, thus, the PROC terminates. The HCI detects that the server has not registered within the allotted time and notifies the client LU that the conversation cannot be started.

HCISERVR abends with a U0100 abend and with a reason code identifying the reason for the abend.

Attached TPs

The HCI can attach TP programs within the HCI address space (no SYSPLEX support). These TPs are ones that required fast startup and that had no need for the allocation services of JCL. In addition, these TPs must be permitted to execute out of an authorized library.

To instruct the HCI to attach a TP, specify the term `ATTACH/ATTACHX` in the first position of the first statement in the PARMLIB member for the TP.

System Security

This section describes the implementation of security required for the HCI Server Activation Facility. Any references to security package nomenclature are for RACF. If RACF is not the z/OS security, the z/OS site needs to make any transpositions to their security package.

The majority of z/OS installations utilize system security. The security requirements are straight forward for the Server Activation Facility process involving submitted batch servers. The workstation user must have the authorization on the z/OS system to do following:

- Execute a job with a unique job name as described in the "Batch Jobs" on page 4-2.

- Execute the module specified in any job step. In some products this could be the batch initiation of TSO using the IBM module IKJEFT01.

The started task system security concerns are more involved. The HCI requires the following three important pieces of data:

- USERID
- OWNER
- GROUP

A started task has none of these when the command is issued and they have to be assigned. This is accomplished by parameters to your site's security package (RACF, ACF/2, or TOPSECRET). Whatever USERID, OWNER, and GROUP is assigned to the started task process must have read or execute access to the HCI authorized library for the execution of the HCISERVR module in the first step of the procedure. This first step in the started task switches the USERID to that of the user, usually at a workstation, making the connection request. After this switch, the started task can access only those resources allowed by the workstation user running under the unchanged OWNER and GROUP. Therefore, the USERID used by the workstation user must have access to any z/OS system and Compuware product libraries designated in those steps following the first step.

For TCP/IP connections, the appropriate USERID and, optionally, GROUP must be obtained by the server TP. The HCI is invoked to set this USERID for the TP instance.

Chapter 5.
Implementing Parallel Sysplex Support

A z/OS sysplex is defined as multiple z/OS images connected together by a hardware (or software) coupling facility (CF). The numerous options available for you to provide this CF support are not discussed here. The HCI uses the cross-system coupling facility (XCF) in order to send and receive messages between components of the HCI when those components are running on different Z/OS images. These systems share all DASD including the security system dataset. A JES2 shared spool environment exists.

The z/OS systems interoperate to create the illusion of a single z/OS. This interoperation required the HCI to implement sysplex support. If you are running a z/OS SYSPLEX, there are no additional requirements to configure z/OS in order to support the HCI.

This chapter describes the different possible configurations for the HCI in a sysplex environment and discusses the different options a customer has when installing the HCI.

HCI Modes of Operation

HCI 3.0 can run in the following execution modes:

- Non-sysplex mode on a non-sysplex system
- Non-sysplex mode on a sysplex system
- Sysplex mode on a sysplex-enabled system.

Non-Sysplex Operational Mode

This section applies to the non-sysplex operation mode of execution on either a non-sysplex or a sysplex z/OS system. Select this mode by specifying **SYSPLEX=**, **PREPROC=**, and **ROUTCMD=NO** or by allowing these attributes to default to these values by omitting them entirely from the HCICNGCA element.

TPs can be submitted (or started) automatically by the HCI, or they may be pre-started by user intervention. These TPs still use the SUBSYSTEM_ID that is specified on the SYSID= attribute of the HCICNGCA element. The HCI does not invoke any of the sysplex code, and it can be run on z/OS systems that do not have sysplex support.

Sysplex Operational Mode

The HCI's sysplex support eliminates the need to run multiple HCI address spaces. One HCI, running on one z/OS image in the sysplex, can control TPs that run on the same or different z/OS images. As a consequence of this single HCI, your site can run a single TCP/IP stack or VTAM address space (on the z/OS that is running the HCI), but TPs can access these communications access methods from any z/OS image.

Multiple HCI address spaces can co-exist in a sysplex, and each can control TPs anywhere in the sysplex. However, these HCIs do not interact with each other — they remain autonomous entities.

The number of z/OS images in the sysplex and the order in which each is activated are arbitrary. Both are handled automatically by the HCI. All of the z/OS images are not required to be active at the time that the HCI is started. As new z/OS images appear in the sysplex, they are automatically added to the HCI, and the preparation job is automatically scheduled to run on these new z/OS systems. If a z/OS image becomes

inactive, it is automatically deleted from the HCI and is re-added when it becomes active once again. The intent is to automate the functions of the HCI as much as possible.

The sysplex mode of operation requires the following three attributes in the HCICNGCA parameters that apply to sysplex execution:

- **SYSPLEX=** Defines a four-character SUBSYSTEM_ID that is used on all z/OS systems in the sysplex. This ID must be different from the SYSID= SUBSYSTEM_ID and must be unique within the sysplex. TPs must use this SUBSYSTEM_ID when registering with the HCI. The inclusion of this attribute causes sysplex support in the HCI to be invoked.
- **PREPROC=** Specifies the name of a procedure (stored in an appropriate PROCLIB) that invokes the HCIYPREP program, which prepares a z/OS system for XCF communication with an HCI.
- **ROUTCMD=** Referenced only when a ROUTE statement in the HCI PARMLIB has no list. **YES** specifies that the HCI routes z/OS START commands to the active member of the sysplex that currently has the least number of routes to it for HCI purposes. **NO** specifies that the HCI invokes started task TPs on the local z/OS system only. The default is NO.

The SYSPLEX= Attribute

Specifying the **SYSPLEX=** attribute on the HCICNGCA element invokes the sysplex processing functions of the HCI. The value of this attribute is a four-character SUBSYSTEM_ID, which must be different from the four-character SUBSYSTEM_ID specified on the SYSID= attribute. Like the SYSID= ID, this SUBSYSTEM_ID must be unique among all the z/OS images in the sysplex and must not be specified in the IEFSSNxx member of SYS1.PARMLIB.

In the sysplex environment two SUBSYSTEM_IDs together define an HCI. The SYSID= name is the local identifier for the HCI. That is, local to the z/OS on which the HCI is running. TPs that run on this z/OS can use this local ID when registering with the HCI. Compuware recommends, however, that TPs not use this name for registration, but use the global name instead. The SYSPLEX= name is the global identifier for the HCI. TPs running anywhere in the sysplex use this name for registering with the HCI, even TPs that are running on the local z/OS image. This recommendation is because the HCI automatically converts the global ID to the local ID when it is discovered that the TP is running on the same z/OS image as the HCI that controls it. This conversion takes place without customer intervention, and ensures that the performance of TPs on the local z/OS does not suffer any degradation caused by the XCF processing. Sites cannot run a TP on the local z/OS and use the sysplex in the HCI.

The PREPROC= Attribute

This PREPROC= attribute specifies the name of the procedure that the HCI invokes on each z/OS image in the sysplex to prepare that z/OS for HCI communication. This user-named procedure must be specified in this attribute. An example of this procedure is shown below:

```
//IEFPROC PROC HCISYSP=DUMY,HCITYPE=
//HCIPREP EXEC PGM=HCIYPREP,PARM='&HCISYSP,&HCITYPE'
//STEPLIB DD DSN=**** APF-authorized load library ****,DISP=SHR
```

This procedure must be available to every z/OS in the sysplex as the name of a START command routed to each z/OS by the HCI. The HCI automatically supplies values for the symbolic parameters that exist in this procedure.

With the exception of specifying the appropriate APF-authorized load library, code this procedure exactly as shown above.

The ROUTCMD= Attribute

When ROUTE is specified in a PARMLIB member without a list, the ROUTCMD= attribute specifies that either **YES**, the HCI is to route z/OS START commands to each z/OS in the sysplex or **NO**, the HCI is to issue z/OS START commands only for the local system. If the ROUTE command is used with a list, only those specified z/OS system names participate in the selection. If yes is specified, the HCI prefixes each START command with *ROUTE sysname*, which causes z/OS to execute the command on the system designated by *sysname*. To determine which z/OS is to run the task, the HCI keeps track of a count of tasks routed to each z/OS. The one with the least number is selected.

The control of which z/OS is to execute submitted jobs is left up to the installation. Appropriate job classes and initiators must be established to ensure that TP jobs submitted by the HCI run on the appropriate system. When **ROUTCMD=YES** is specified, **JESJPRM=NO** should also be specified so that TPs submitted by the HCI are allowed to execute on any z/OS image in the sysplex. See "<HCICNGCA Element" on page 9-4 for a complete description of these attributes.

Installation on a Sysplex

Complete the following steps to implement sysplex support:

1. Ensure that the HCI load libraries are APF-authorized and available on all z/OS images. They must be APF-authorized on every z/OS, not just on the z/OS where the HCI runs.
2. Install a new procedure in a PROCLIB that is either shared on all z/OS images installed in a PROCLIB on each z/OS image. The PROCLIB must be one from which started tasks can be defined. This procedure invokes the HCIYPREP program, which prepares the z/OS for HCI sysplex execution. You must specify the user-named procedure on the **PREPROC=** attribute of the HCICNGCA element. See "The PREPROC= Attribute" on page 5-2 for more information about this attribute.
3. Specify in the **SYSPLEX=** attribute of the HCICNGCA element a new, unique four-character SUBSYSTEM_ID that is different from any other SUBSYSTEM_ID in use on any of the z/OS images. Just like the SYSID=SUBSYSTEM_ID, this name must not be included in the IEFSSNxx member of SYS1.PARMLIB.
4. Change the word *START* to the word *ROUTE* in any HCI PARMLIB member that defines a TP that can be started on any z/OS in the sysplex. Further, if it can run on only one or more systems, but not all, specify the systems in a list. See "ROUTE(System_List)" on page 4-2 for more information.
5. If the HCI is to automatically start *started task* TPs, you must decide whether these TPs are to be started only on the z/OS on which the HCI is running or can they execute on any z/OS in the sysplex when only ROUTE without a list is specified. Specify **ROUTCMD=YES** if any z/OS can run these TPs. Specify **ROUTCMD=NO** if TPs must run on the HCI's z/OS image.

   Note: If you specify **ROUTCMD=YES** but the system is not sysplex-enabled, it is treated as ROUTCMD=NO.

HCIYPREP Program

The HCIYPREP program initializes a z/OS such that HCI TPs can run on that z/OS and can be controlled by an HCI on another (or the same) z/OS image. This program loads executable modules into storage, establishes a subsystem, and a z/OS/PC routine. When HCIYPREP terminates, these elements remain available to HCI TPs.

HCIYPREP is started automatically on each z/OS in the sysplex when the following occurs:

1. The HCI starts.
2. A new z/OS is added to the sysplex.
3. An old z/OS that left the sysplex has subsequently rejoined it.

Under normal circumstances, an operator should *not* have to intervene and run this program. When HCIYPREP has completed its initialization successfully, it signals this fact to the HCI, which then marks the associated z/OS as ready to process TPs. Until HCIYPREP ends normally, the z/OS is considered to be in a pending active state, and TPs are not started on this z/OS. A warning message indicating which z/OS images are in this pending state is written in two minutes of the attempted start and then every hour until started.

If HCIYPREP did not run successfully, it may have to be started manually after the problem it encountered has been fixed. The operator should issue the same z/OS START command as the HCI issued, which can be found on the z/OS SYSLOG.

Shared DASD Considerations

Although it is the purpose of a SYSPLEX to have two or more independent Z/OS images tightly coupled, it does not necessitate total separation. In order to ease the maintenance issues, shared DASD should be utilized between the members of the SYSPLEX. One candidate for the shared DASD space should be the HCI authorized loadlib data set. This is particularly true if the HCI Server Activation Facility routes started tasks to other members of the SYSPLEX. These routed started tasks require access to the HCI authorized library.

Automatic Restart Management

The HCI supports z/OS Automatic Restart Management (ARM) facility as documented in the IBM manual z/OS VnnRnn MVS Sysplex Services Reference, document number SA22-7618. A thorough understanding of parallel sysplex and of ARM support is required before implementing it in an HCI instance.

Requesting Automatic Restart Management Support

ARM support is requested by including ARMNAME="xxxx.....xxxx" XML element or ARMNAME= macro operand in the HCICNGCA element or macro. Additionally an option element ARMTYPE="xxxxxxxx" or ARMTYPE=xxxxxxxx if your installation wishes to specify the system defined TYPE for the named element.

Your installation may wish to specify particular processing parameters to be obeyed by the system when an automatic restart occurs, or you may wish to let the system defaults take effect.

VTAM Generic Resources

VTAM's Generic Resource facility uses services of the MVS Coupling Facility in order to maintain a list of generic names and the association of each generic name to a specific name (APPLID). Remote LUs desiring to establish LU 6.2 or LU 2 sessions with the HCI can do so by specifying the generic name in place of the actual (APPLID) name. When VTAM receives a session request directed to a generic name, it assigns the session to one of the applications that has registered with the generic name. In this way, multiple HCIs can run in a sysplex environment, one on each MVS image, for instance, and VTAM ensures that the session load is balanced among the various HCI instances. In addition, each HCI can support multiple LU 6.2 and/or LU 2 ACBs, each of which can register with

the same or different generic name. In this environment, multiple LU 6.2 and/or LU 2 sessions can better utilize the CPU resources by being separated onto different ACBs.

Requesting Generic Resource Support

Request Generic Resource support by adding an attribute to the HCICNAMB element named GENNAME=*xxxxxxx*, where *xxxxxxxx* is the generic name that the associated ACB is to register with. Multiple HCICNAMB elements can specify the same generic name, but all HCICNAMB elements that do specify the same generic name must be of the same LU type (LU62 or LU2). HCICNAMB elements in different HCIs can specify the same generic name. ACBNAME, however, must *always* be unique within the network.

Operational Considerations

For the HCI successfully to register a generic name, the coupling facility must be installed and must be activated. If it is not, the registration process fails, a message is displayed on the master console, and HCI initialization continues without the generic name. If this failure occurs, the HCI requests that MVS notify it whenever the status of the coupling facility changes. When this notification occurs, the HCI retries the request to create the generic name. This process continues until the generic name is successfully added. Thus, it is not necessary to stop and restart the HCI for it to begin to honor generic names.

Chapter 6.
Using the Journaling Facility

This chapter describes the HCI's extensive journaling (logging) facility. Every online system requires a method for recording what has occurred within that system in order that problems can be researched later.

The journaling facility is a separate task within the HCI. This task is responsible for writing records to the current journal dataset. Additionally, this task is responsible for allocating, deallocating, opening, and closing the journal datasets as necessary.

Any number of individual journal datasets can be configured into the HCI although only one of them is ever allocated and opened at a time. Each of these is a VSAM ESDS, and is usually defined with no secondary extents.

One journal is active at a time. The first one specified in the configuration member (see "<HCICNJCB Element" on page 9-17) is dynamically allocated and opened. Journal records are written to this dataset. When it fills, or upon an operator command, this dataset is closed and dynamically deallocated. The next dataset in the list is then allocated and opened, and journaling continues.

When the last journal in the list has been filled, the first one is again allocated and opened. In order to preserve journal data, ensure that the specified number and size of journal datasets is large enough to allow journaling to continue while the batch journal print or journal unload is run.

Allocating and Initializing Journal Datasets

Figure 6-1 shows an example of executing IDCAMS to create a journal data set. Replicate the control statements in this example to create more than one VSAM ESDS.

Figure 6-1. JCL to Allocate Journal Datasets

```
//*
//* place your job card here
//*
//DEFINE EXEC PGM=IDCAMS,REGION=3M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
     DELETE HCI.VSAM.JOURNAL CLUSTER
     DEFINE CLUSTER (NAME(HCI.VSAM.JOURNAL)   -
     TRK(270) VOL(??????)                     -
     CISZ(16384) SPANNED                      -
     BUFSP(163840)                            -
     RECSZ(60 32600) NONINDEXED REUSE         -
     SHR(1 3) SPEED)
/*
```

Initialize the journal whenever its contents are no longer of interest. JCL to initialize a journal dataset follows:

Figure 6-2. JCL to Initialize Journal Datasets

```
//*
//* place your job card here
//*
//INIT  EXEC PGM=IDCAMS,REGION=3M
//SYSPRINT DD SYSOUT=*
//DUMMY    DD DUMMY,DCB=(RECFM=VB,LRECL=32604,BLKSIZE=32608)
//JOURNAL  DD DSN=HCI.VSAM.JOURNAL,DISP=SHR
//SYSIN    DD  *
   REPRO INFILE(DUMMY) OUTFILE(JOURNAL)
/*
```

Replace the DSN for the journal by the name used at your installation.

Note: The Compuware Workbench Host Communication Configuration Facility (HCCF) automates the coding of the <HCICNJCB elements and the generation of the JCL to allocate and initialize the HCI journals. See the *Compuware Workbench Installation Guide* for more information.

Chapter 7. Operator Communications

Messages to the operator are displayed from the HCI via the standard HCI message processing facility. All error and informational messages are defined in a dataset that is processed by the HCI message processing system. One or more load modules are created that are used to generate operator messages. Variable text is provided by the HCI in the calls to display error messages.

You can issue commands via the z/OS operator console interface.

Operator Commands

You can enter operator commands to the HCI via a Write to Operator Request (WTOR) or via the z/OS MODIFY, depending on the value specified for the configuration HCICNGCA element's OPCMD attribute. See "OPCMD="WTOR|modify"" on page 9-9 for more information.

Operator commands let you deactivate the HCI region, display information about the HCI region, and manipulate the HCI journals.

Commands List

All supported operator commands are listed below. The uppercase letters of each command are the minimum characters required to execute the valid short form of the command.

- ACTivate subtask
- DEactivate subtask
- Display
- Open
- Close
- JMask
- REFRESH (modname)
- REFRESH XML (modname)
- SDump
- SHUTDOWN
- SWap

Commands Syntax

Separate the command from the attributes by one or more spaces (no comma). If you specify more than one attribute, separate each from the next with a comma (no spaces). Not all commands require attributes.

ACTIVATE subtask

The ACTIVATE command restarts a VTAM- or TCP/IP-related subtask that has previously been deactivated either by operator command or by exhausting the retry count in the subtask recovery routine.

DEACTIVATE subtask

The DEACTIVATE command directs the HCI to deactivate the subtask named in the command. Attributes for this command include normal deactivation and a forced abend deactivation.

DISPLAY

The DISPLAY command allows the operator to display upon his console various components and statuses within the HCI. Table 7-1 shows the available attributes for the DISPLAY command.

Table 7-1. DISPLAY Command Attributes

Attribute	Description
ALL	Displays all HCI status fields.
JOURNAL	Displays information about the journal.
STOR	Displays current virtual storage utilization.
CB	Displays current control block utilization.

OPEN

The OPEN command directs the HCI to allocate and open the first journal dataset and begin journaling.

CLOSE

The CLOSE command directs the HCI to close and deallocate the current journal dataset and to refrain from any further journaling until an OPEN command is issued.

JMASK

The JMASK command sets the journal mask to on or off. Table 7-2 shows the available attributes for the JMASK command.

Table 7-2. JMASK Command Attributes

Attribute	Description
ON	Sets the journal mask to all ones (x'FFFFFFFFFFFFFFFF').
OFF	Sets the journal mask to zeroes (x'0000000000000000').

REFRESH =modname (*HCI 2.5 only)*

The REFRESH command is supported only by HCI Release 2.5 and has been removed from HCI Release 3.0. This command directs the HCI to read the configuration module named *modname* from the //FDBDRPL DD concatenation and to dynamically update the running parameters in the HCI from the values in the named configuration module. If modname is omitted, the HCICNFIG element is used as the default.

REFRESH XML=xmlname (*HCI 3.0 only*)

The refresh command directs the HCI to read the specified XML document member from the //HCIXML DD and to dynamically apply the values contained in this document to the running HCI. Note that for the dynamic refresh capability for XML to be active, the HCIXML DD must refer to a partitioned data set. The xmlname operand on this command specifies the member name within this PDS which is to be parsed and applied to the running HCI (see “Dynamic Refresh Facility” on page 9-1).

SDUMP

The SDUMP command prepares z/OS and the HCI to take an SDUMP. This command sets the four internal values of the HCI as if the attributes DMPPFX, DMPVOL, DMPCLAS, and DMPUNIT of the HCICNGCA configuration element had been set to the appropriate values for an SDUMP. In order to have the HCI take an SDUMP, the DMPUNIT must be set to a valid SYSUNIT name, and the DMPPFX must be set to SDUMP. The others, DMPVOL and DMPCLAS, must not have been specified.

The SDUMP command resets the DMPVOL and DMPCLAS internal values. The two positional attributes—SYSUNIT and CONNAME (console name) separated by a comma (,)—are optional. If the second is specified, the comma must be present. If the second attribute is omitted, those operator commands to set the options for the SDUMP are displayed only. If the second attribute is specified, the HCI issues the operator commands to establish the proper SDUMP options for an HCI SDUMP. The original values can be reset only by restarting the HCI or issuing the REFRESH command.

Table 7-3 shows valid SDUMP command syntax variations:

Table 7-3. SDUMP Command Syntax

Command Syntax	Description
SDump	Displays the operator commands to set the SDUMP options. This form does not change any of the HCI internal values.
SDump sysunit	Sets the DMPUNIT internal value and displays the operator commands to set the SDUMP options.
SDump sysunit,conname	Sets the DMPUNIT internal value and issues the console commands to correctly set the SDUMP options.
SDump ,	Sets SYSDA as the internal DMPUNIT name and displays the operator commands to set the SDUMP options.
SDump *,conname	Uses the currently set DMPUNIT internal value and issues the operator commands to set the SDUMP options.
SDUMP ,conname	Displays current control block utilization.

SHUTDOWN

The SHUTDOWN command directs the HCI to terminate. Table 7-4 shows the available attributes for the SHUTDOWN command:

Table 7-4. SHUTDOWN Command Attributes

Attribute	Description
IMMED	Causes an immediate shutdown.
NORMAL	Causes a normal shutdown
ABEND	Causes a forced abend.

If none of the attributes is specified, a normal shutdown occurs.

SWAP

The SWAP command directs the HCI to close and deallocate the current journal dataset, and to allocate and open the next one.

Chapter 8.
HCI Diagnosis and Debugging

If problems occur with the HCI, Compuware may ask its customers to provide the following information that aids in diagnosing HCI problems:

- Storage dumps
- HCI journal
- Stub tracing
- Generalized Trace Facility (GTF) trace.

This chapter discusses these information types.

Storage Dumps

The HCI invokes the z/OS dumping facilities whenever it detects a problem that cannot automatically be corrected. HCI can request two major types of dumps: SNAP dumps and SDUMPs.

SNAP dumps can be directed to a disk dataset or to SYSOUT, based on the DMPCLAS, DMPPFX, DMPUNIT, and DMPVOL attributes on the HCICNGCA element. Specifying DMPCLAS makes specifying the other DMPxxx attributes unnecessary. Specifying DMPPFX, DMPUNIT, and DMPVOL makes specifying DMPCLAS unnecessary. SNAP dumps are relatively slow to generate; consequently, HCI execution may be suspended for an unacceptably long length of time.

SDUMPs are always directed to the installation-defined SYS1.DUMPxx (or equivalent) datasets. These dumps can be managed by the installation more easily than SNAP dumps can be, and they can be copied easily to disk or tape. These dumps are generated in a very short length of time, thus suspending HCI execution for a minimal duration. Request an SDUMP by coding the DMPPFX=SDUMP attribute (and omitting DMPCLAS and DMPVOL attributes) on the HCICNGCA element. Compuware strongly recommends that you code DMPPFX=SDUMP at your site.

Ensure that any dumps sent to Compuware contain the following information:

- Abending PSW
- Abending register contents
- Storage: region, private, common, LPA, SQA, and LSQA
- Region system control blocks, such as ASCB, TCBs, IRBs, among others.
- Global resource enqueues
- System trace tables
- Dump summary.

If you are generating an SDUMP, specify the following options:

```
RGN,PSA,CSA,LPA,SQA,LSQA,GRSQ,TRT,SUM
```

and

```
TYPE=XMEME
```

If the HCI is running in a sysplex and one or more of the TP applications are running on a different z/OS image than the HCI, then also specify the following option:

```
COUPLE
```

HCI Journal

The HCI journal provides the most comprehensive debugging aid available. Although not required for most HCI execution, when a problem occurs, the journal should be created that contains the execution error. The journal facility is always available in the HCI, but based upon the setting of the journal mask attribute in the HCICNGCA element and the availability of journal VSAM datasets, the journal may not be active at any given time.

If you need to provide Compuware with the journal data for analysis, prepare it for FTP transmission.

CAUTION:
Do not send a print image of the journal data to Compuware. Doing so will only result in a delay to the problem analysis. For more information about FTP transmission, see "Preparing Journal Data for FTP Transmission" below and TCP/IP.

Preparing Journal Data for FTP Transmission

Reformat the contents of a journal dataset into a dataset that is appropriate for FTP transmission to Compuware's FTP site. Figure 8-1 shows an example of the JCL to perform this reformat.

Figure 8-1. JCL to Reformat Journal Data

```
//*
//* PLACE YOUR JOB CARD HERE
//*
//UNLOAD EXEC PGM=HCIJUNLD
//STEPLIB  DD DSN=users.authorized.hci.loadlib,DISP=SHR
//HCIINDD  DD DSN=HCI.JOURNAL.DATA,DISP=SHR
//HCIOUTDD DD DSN=users.new.journal,DISP=(,CATLG),
//            UNIT=SYSDA,SPACE=(TRK,(15,15),RLSE,CONTIG)
//SYSUDUMP DD SYSOUT=*
```

Once the data has been successfully unloaded, the unloaded dataset can be transmitted via FTP to Compuware's FTP site. See "Using the TCP/IP FTP Program" on page 8-5 for more information.

Printing the Journal Data

Do not send a print image of the journal data be sent to Compuware because doing so will delay problem analysis. However, if you do need to print the journal data, use the JCL shown in Figure 8-2 to minimally format the journal data.

Figure 8-2. JCL to Print Journal Data

```
//*
//* PLACE YOUR JOB CARD HERE
//*
//JRNLPRT EXEC PGM=HCIJRPRT,REGION=32M
//STEPLIB DD DSN=users.authorized.hci.loadlib,DISP=SHR
//*
//LOGIN DD DSN=HCI.JOURNAL.DATA,DISP=SHR
//SYSPRINT DD SYSOUT=*
//LOGPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

GTF Trace

Instances for which a Generalized Trace Facility (GTF) trace may be required include:

- VTAM trace
- TCP/IP trace
- Sysplex trace.

All aspects of preparing, starting and stopping the GTF, as well as sending the GTF data remain the same. The only difference is the GTF parameters.

In an active sysplex environment, you cannot use the HCI journal facility to record the execution of the z/OS systems other than the local z/OS that contains the HCI itself. Because some sort of journal is required, use the z/OS GTF to record sysplex processing.

Preparing GTF to Gather HCI Data

Do the following to prepare to use the GTF to gather HCI data:

1. Allocate a dataset to contain the GTF trace output.
2. Create a GTF procedure to invoke GTF with the dataset.
3. Create a SYS1.PARMLIB member to contain the GTF execution time parameters.

Figure 8-3 shows an example of a job that allocates a dataset to receive the GTF trace data.

Figure 8-3. JCL to Allocate a GTF Trace Dataset

```
//*
//* PLACE YOUR JOB CARD HERE
//*

//ALLOC   EXEC PGM=IEFBR14
//GTFDATA   DD DSN=HCI.GTFTRACE.DATA,DISP=(,CATLG),UNIT=SYSDA,
//             SPACE=(CYL,(40),,CONTIG),VOL=SER=volser,
//             DCB=(RECFM=VB,LRECL=8232,BLKSIZE=8236,DSORG=PS)
```

Specify a dataset name and volume serial number appropriate for your site's installation. Code the DCB parameters exactly as shown.

Figure 8-4 shows an example of a procedure to invoke GTF.

Figure 8-4. Procedure to Invoke GTF

```
//GTFHCI  PROC MEMBER=GTFHCI
//IEFPROC EXEC PGM=AHLGTF,PARM='MODE=EXT,DEBUG=NO,TIME=YES',
//             REGION=2280K,DPRTY=(15,15)
//IEFRDER   DD DSNAME=HCI.GTFTRACE.DATA,DISP=SHR
//SYSLIB    DD DSNAME=SYS1.PARMLIB(&MEMBER),DISP=SHR
```

The following is an example of a SYS1.PARMLIB member associated with the GTF procedure.

```
TRACE=USRP
USR=(FE1,FE2,FE4,FEF,FF0,FF1,FF2)
END
```

Starting GTF

Start the GTF procedure by entering a command on the system console or from an SDSF panel. The following is an example of this command.

```
START GTFHCI.GTF
```

The GTF program displays startup information, including a listing of the parameters that are to be used. If its initialization is successful, GTF requests that the operator confirm the startup parameters by replying to an outstanding WTOR. This message is shown below:

```
*nn AHL125A RESPECIFY TRACE OPTIONS OR REPLY U
```

Where *nn* is the reply number. Using the reply number *nn*, the operator enters the following on the system console or on an SDSF panel:

```
Rnn,U
```

GTF displays more diagnostic information, ending with the following message:

```
AHL031I GTF INITIALIZATION COMPLETE
```

Note that when GTF has successfully been started, actual tracing to the GTF dataset does not occur until the next TP registers with the HCI. Thus, in-flight conversations are not be traced. Compuware recommends starting GTF before starting any HCI TPs. In this way, all information concerning those TPs is traced.

Stopping GTF

Stop GTF by entering the following command on the system console or from an SDSF panel:

```
P GTF
```

GTF responds with the following message:

```
AHL006I GTF ACKNOWLEDGES STOP COMMAND
```

Sending GTF Data for Analysis

You can reformat the contents of the GTF dataset into a dataset that is appropriate for FTP transmission to Compuware's FTP site or placed onto magnetic tape to be mailed. Refer to one of the two applicable sections.

Preparing GTF Data for FTP Transmission

In order to send the GTF data via FTP, reformat the data. Figure 8-5 shows an example of the JCL to perform this reformat.

Figure 8-5. JCL to Reformat GTF Data

```
//*
//* PLACE YOUR JOB CARD HERE
//*
//UNLOAD EXEC PGM=HCIJUNLD
//STEPLIB  DD DSN=users.authorized.hci.loadlib,DISP=SHR
//HCIINDD  DD DSN=HCI.GTFTRACE.DATA,DISP=SHR
//HCIOUTDD DD DSN=users.new.gtfdata,DISP=(,CATLG),
//            UNIT=SYSDA,SPACE=(TRK,(15,15),RLSE,CONTIG)
//SYSUDUMP DD SYSOUT=*
```

Once the GTF data has been successfully unloaded, you can transmit the unloaded dataset via FTP to Compuware's FTP site.

Using the TCP/IP FTP Program

The programs HCIJUNLD and HCIGUNLD are designed to read the variable length input data and create a fixed length output dataset that can be transmitted by File Transfer Protocol (FTP). The format of the records created by HCIJUNLD and HCIGUNLD is known by the corresponding programs that re-create the original datasets.

Invoking FTP

If TCP/IP is available on a z/OS host, FTP can be invoked from a TSO session, with or without ISPF. If TCP/IP is not available, the output of HCIJUNLD or HCIGUNLD must be transmitted to a workstation that does have TCP/IP on it before the dataset can be sent to Compuware.

The remainder of this section assumes FTP is available on the host.

Assuming a host-based TCP/IP and no fire wall, enter the commands shown in Figure 8-6 on page 8-6 on any ISPF screen. Commands are shown with a right arrow (===>) prefix.

Figure 8-6. Example of FTP Session with Compuware

```
===> TSO FTP FTP.COMPUWARE.COM
Using 'SYS1.TCPPARMS(FTPDATA)' for local site configuration parameters.
IBM FTP CS V1R11
Connecting to: ftp.compuware.com 192.168.97.200 port: 21.
220 Please contact ftp@compuware.com if you have any problems or questions.
NAME (ftp.compuware.com):

===> ANONYMOUS
>>>User anonymous
331 Password required for anonymous.
PASSWORD:

===> USERID@CUSTOMER.COM
>>>PASS
230 Login OK. Proceed.
Command:

===> CD PUB/CSS/INCOMING
>>>CWD pub/css/incoming
250 Folder changed to "/pub/css/incoming".
Command:

===> BIN
>>>TYPE I
200 Type set to I
Command:

===> PUT HCI.OBJECT(HCI30) HCI30.BIN
>>>SITE FIXrecfm 41600 LRECL=4160 RECFM=F BLKSIZE=4160
501 Command not understood.  (You can ignore this error.)
>>>PORT 10,10,0,204,167,33
200 Command okay.
>>>STOR hci30.txt
150 Opening BINARY mode data connection for hci30.bin.
226 Transfer complete. 12720 bytes transferred. 12720 bps.
12720 bytes transferred in 0.005 seconds. Transfer rate 2544.00 Kbytes/sec
Command:

===> QUIT
221 Service closing control connection.
```

Storage Estimates

The storage utilized by the HCI is very use-oriented. The type of storage that is being used can be categorized into four types each above and below the line:

- Private
- Common
- Module
- Buffer

Private and common storage can each be more than 1 MB above the line. The storage used below the line for private and common is minimal at around 20 KB. Module storage is under 700 KB, the majority of it above the line (less than 50 KB below the line). Buffer storage can be less than 100 KB above the line. Above the line storage is always used unless unavailable or not permitted to be used by some of the very few z/OS facilities. No storage is allocated above the 64-bit bar.

These amounts are all rough estimates and the actual storage size is directly related to the functions the HCI is performing. The actual storage utilization can be seen using the HCILOOK facility or displayed on the operator console using the **DISPLAY STOR** HCI command.

To help explain where some of the storage is used, Table 8-1 shows those control blocks whose quantity is user-controlled:

Table 8-1. User-Controlled Control Blocks Storage

Control Blocks	Description
AMB	Access Method Block 728 CSA below
CCB	Conversation Control Block 1264 Private above
DCB	Destination Control Block 760 Private above
JCB	Journal Control Block 104 Private above
PCB	TCP/IP Port Control Block 488 Private above
RCB	Region Control Block 96 CSA above
TPT	TP Profile Table 320 Private above
UIB	User Interface Block 1040 CSA above
WRE	Work Request Element 256 CSA above

The WREs are usually the largest storage item. Currently the HCI is intolerant of running out of most internal control blocks. Always provide a few extra of each where possible and monitor storage utilization via the TSO/ISPF HCILOOK facility.

Chapter 9.
Using the Source Parameter Facility

The Source Parameter Facility (SPF) is new with HCI 3.0 and provides a means for supplying parameters to the HCI which does not require the two-step assembly and link edit of a configuration load module. Support for the load module remains within HCI 3.0, and the parameter macros are still shipped with the HCI. Compuware recommends that you follow the steps described in “Migration Procedure” on page 2-2 at some time after HCI 3.0 has been installed.

The Source Parameter Facility provides a much more complete diagnostic feature whereby errors in the parameter specifications can be repaired, in most cases, without contacting Compuware for help. See the //HCIERR DD statement description in “New DD Statements” on page 2-1. This DD statement defines the dataset containing any error messages generated by the HCI concerning the parameter specifications.

The source parameters reside in a single XML document that contains multiple elements, each of which contains multiple attribute names and values. See Appendix A, “Sample XML Parameters” for an example of a parameter document.

See “Migration Procedure” on page 2-2 regarding how you can create the XML document initially. Compuware has ensured that the XML created for migration matches the parameters initially specified by the set of macro attributes.

The HCIXML DD statement refers to a sequential data set or to a partitioned data set (with a member name). The sequential data set option is available only for migration purposes and Compuware recommends you allocate a PDS with DCB=(RECFM=FB,LRECL=80,BLKSIZE=nnnn,DSORG=PO) with primary and secondary extents and enough directory blocks that you will not run out of them. Into this PDS you must place the XML document which contains the initial execution parameters (see The XML Document next in this section). You may name this member anything you like and specify this name on the member name associated with the DSN= operand on the HCIXML DD statement. In addition, you may place XML documents which are to be dynamically applied to the running HCI via the REFRESH XML= console command. The member name of the XML document to be used for the refresh is specified in the command, (for example: REFRESH XML=NEWCNFIG). These refresh documents are not processed unless and until you issue the REFRESH XML= command.

Dynamic Refresh Facility

Dynamic refresh is available whether you are using the two-step macro assemble/bind configuration member or the source XML facility for configuring the HCI. In either case it is possible to create a new configuration module/document and request that the HCI merge the information in the new set of parameters with those that are currently running in the HCI.

You may make changes to the original module/document and use it again for the refresh. Any unchanged parameters will remain in effect. New or changed ones will replace or add to those in effect at the time of the refresh.

Compuware suggests that you make a copy of your running parameters and make changes to the copy and use the copy for the refresh. In this way, any errors encountered during the refresh can be backed out by refreshing the original module/document again.

In the following section you will find every XML element and attribute which are supported in the initial and refresh module/document. If you are adding new AMB, PCB,

JCB, SIT or TPTs, then every attribute is available to you just as it would be if you were starting the HCI from scratch. If you are updating an existing copy of one of these control blocks, then only those attributes which specify "Refresh: Yes" can be changed.

Since you may have only one HCICNGCA element, you can change only those attributes which indicate that they can be changed during a refresh.

The refresh document must be a well-formed XML document (or an assembler/bound load module with no assembly or binder errors). If this is not the case, the refresh will terminate and no changes to the running system will occur.

The XML Document

The HCI XML document is composed of one root element and at least four subordinate elements.

Root Element

The root element for the configuration document consists of <HCICNFIG> as the first statement in the dataset and </HCICNFIG> as the last statement. All other configuration parameters reside within these tags:

```
<HCICNFIG>
... remaining parameters ...
</HCICNFIG>
```

Required for initial and refresh documents.

Subordinate Elements

Six distinct element names can be specified with the starting and ending root element tags:

- <HCICNGCA
- <HCICNAMB
- <HCICNPCB
- <HCICNJCB (optional)
- <HCICNSIT (optional)
- <HCICNTPT

Each of these elements is terminated by a single /> ending tag. Note that you can omit the tag name.

Only one HCICNGCA element can be specified in the document. All other elements can be repeated as many times as necessary to complete the description of the HCI execution and connectivity.

Figure 9-1 shows an example of a parameter document without any attributes specified.

Figure 9-1. Parameter Document Without Attributes Example

```
<HCICNFIG>
  <HCICNGCA
   ...attributes for HCICNGCA...
  />
  <HCICNAMB
   ...attributes for HCICNAMB...
  />
  <HCICNPCB
   ...attributes for HCICNPCB...
  />
  <HCICNJCB
   ...attributes for HCICNJCB...
  />
  <HCICNSIT
   ...attributes for HCICNSIT...
  />
  <HCICNTPT>
   ...attributes for HCICNTPT...
  />
</HCICNFIG>
```

Notes:

1. One and only one HCICNGCA element must be present.
2. All other elements can be replicated as many times as necessary. All HCICNAMBs must be together, followed by all HCICNPCBs, followed by all HCICNJCBs, then HCICNSITs, and finally HCICNTPTs.
3. You may not have any HCICNJCB or HCICNSIT elements, but you must have one HCICNGCA and at least one HCICNAMB, one HCICNPCB, and one HCICNTPT.
4. Comments, which are stripped out before passing the document to the parser, are allowed by inserting an asterisk (*) in position 1 of any statement.
5. XML document comments (<!--comment text-->) are allowed only between elements; never within them.
6. Comments are not be listed in either the HCIPRINT nor HCIERR datasets.
7. Positions 1 through 72 can contain parameter information. Positions 73 through 80 are ignored and can contain sequence numbers if desired.
8. If an attribute value must span a statement, code the value up through position 72 of the first statement, and then begin in position 1 of the next statement. Enclose the entire value in double quotes. Be careful when continuing values and using the system variable substitution feature because unexpected results can occur.
9. All attribute values must be enclosed in double quotes.
10. All elements are supported but optional for refresh documents as indicated later when each attribute element is discussed.

Relationships Between Elements

The following relationships must exist between XML elements:

- For each HCICNSIT element that specifies a TCPNAME, there must be a corresponding HCICNAMB specifying the same TCPNAME.
- For each HCICNSIT element that specifies a PORT number, there must be a corresponding HCICNPCB specifying the same PORT number.

- For each HCICNPCB element that specifies a TPNAME, there must be a corresponding HCICNTPT specifying the same TPNAME.
- For each HCICNPCB element that specifies a TPNAME, there must be a corresponding HCICNAMB specifying the same TPNAME.
- For each HCICNTPT element that specifies an ALIASOF attribute, there must be a corresponding HCICNTPT with the ALIASOF as its TPNAME.

<HCICNFIG> Element

The <HCICNFIG> element is the root element for the XML document. All XML statements reside between this element and the corresponding closing element. This element has no attributes. Therefore, unlike the subordinate elements, code it with a right angle bracket (>) immediately following the element name. This element is required for initial and refresh documents.

<HCICNGCA Element

One and only one <HCICNGCA element must be in the parameter document. Note that the element name is not closed immediately, but remains open until after all attribute names and values for this element have been coded.

The following attribute names are valid within the HCICNGCA element. Note that each attribute is shown below with its default value. The HCI migration facility will insert the actual values when it creates the initial document. You may retain these values, change them, or delete them entirely.

All attribute names are allowed in the initial configuration document, however only some of the attributes are allowed in a refresh document (see the <HCICNGCA element attribute description below for more information).

<HCICNGCA Attribute List

SYSID="sysid"

Required but can be overridden by EXEC PARM.

Default: None

Refresh: No

SYSID is the short form for SUB_SYSTEM_ID and must be specified either here in the XML parameter document or on the PARM= operand of the EXEC statement that invokes the HCI. See Chapter 3, “Executing the HCI” for more information. This value must be four characters in length and is used by the HCI initialization routines to create a z/OS subsystem. You must ensure that this four-character value is unique among all subsystems running on this LPAR, and you must *not* define this value in SYS1.PARMLIB. The HCI dynamically creates this subsystem for you. Each HCI that you desire to run on a single LPAR must have its own unique SYSID value.

If you specify the SYSID attribute in the XML document and on the PARM= attribute of the EXEC statement, the EXEC statement PARM overrides the SYSID specified here if they differ.

ARMNAME="xxxx...xxxx"

Optional

Default: ARMNAME=""

Refresh: Yes

The Automatic Restart Management Facility in the HCI is activated by including this element and specifying a non-null name.

- Uppercase alphabetic characters
- The numbers 0 through 9
- $ # @ and underscore (_)

The first character may not be a number.

The name must be 16 characters long, padded on the right with blanks.

The name must be unique across the sysplex.

Element names that begin with **A** through **I** and **SYS** are reserved for use by IBM.

ARMTYPE="xxxxxxxx"

Optional

Default: ARMTYPE=""

Refresh: Yes

If ARMNAME is specified and if your installation requires that an element type be specified in the IXCARM TYPE=REGISTER macro, then specify the type here

- Uppercase alphabetic characters
- The numbers 0 through 9
- $ # @ and underscore (_)

The first character may not be a number.

The name must be 8 characters long, padded on the right with blanks.

The name does not have to be unique.

Element types that start with **A** through **I** and **SYS** are reserved for use by IBM.

DAE="NO|yes"

Optional.

Default: DAE="**NO**"

Refresh: Yes

"NO"

The z/OS Dump Analysis and Elimination (DAE) facility can be detrimental in obtaining dumps of the HCI. Specifying **DAE="NO"** or allowing this attribute to default to "NO" causes the HCI to change portions of the summary dump in such a way that DAE never suppresses an HCI dump.

"YES"

Specifying **DAE="YES"** means that the DAE facility is to suppress what it considers duplicate dumps.

DFLTUSR="userid"

Optional.

Default: DFLTUSR="**HCIUSER**"

Refresh: Yes

Specifies the name of a valid RACF (ACF/2 or TOPSECRET) user ID to be used by the HCI when TP jobs are submitted or started and when there is no other user ID available.

This user ID must be assigned within the security system and must be granted authority to perform whatever processing the TP requires up to the point in time

when the TP issues one of the set security calls. Until that time, the new user ID is in effect. For example, ensure that the default user ID has either ACCESS(READ) or ACCESS(EXECUTE) to the //STEPLIB DD used in the TP JCL and that it has appropriate access to any other datasets that it processes under the default user ID.

Once the TP has issued a security call, the default user ID is no longer used for access because the TP starts running under the user ID passed in the security call.

DMPCLAS="class"

Optional.

Default: None

Refresh: Yes

Specifies a valid JES/n output class to which SNAP dumps are to be directed. If this attribute is omitted, then the DMPPFX, DMPUNIT and DMPVOL attributes define the output dataset to contain SNAP dump information.

DMPPFX="SDUMP|prefix"

Optional.

Default: DMPPFX="**SDUMP**"

Refresh: Yes

"SDUMP"

Specifies that all dumps taken by the HCI are system dumps (via SDUMPX) and are directed to the SYS1.DUMPxx (or customer-defined) dataset. The DMPUNIT, DMPVOL, and DMPCLASS attributes are ignored.

"**prefix**"

Causes a dataset to be allocated whose DSNAME is prefixed by this prefix and a SNAP dump is taken to this dataset. In this case, the DMPUNIT and DMPVOL attributes further define the location of the dataset. The prefix can be from 1 to 12 valid characters for a DSNAME prefix. Note that the user ID under which the HCI is executing must have ACCESS(ALTER) to this DSNAME prefix.

DMPUNIT="unitname"

Optional.

Default: None

Refresh: Yes

Specifies the unitname for SNAP dumps allocated by the HCI. If this attribute is omitted, then the DMPPFX attribute must specify "**SDUMP**". *Unitname* must be a valid name for DASD allocation at your installation; for example, DMPUNIT="**SYSDA**".

DMPVOL="volser"

Optional.

Default: None

Refresh: Yes

Specifies the VOLSER for SNAP dumps allocated by the HCI. If this attribute is omitted, then the DMPPFX attribute must specify "**SDUMP**". VOLSER must be a valid volume serial number for DASD allocation at your installation.

EOM="YES|no"

Optional.

Default: EOM="**YES**"

Refresh: Yes

Specifies whether the HCI subsystem is to monitor the system for end of memory (EOM) events.

"YES"

End of task events, which are always monitored, will provide the necessary protection against TP program ending without the HCI being aware of the termination. The default exists due to downward compatibility requirements.

"**NO**"

Causes less CPU overhead by the HCI.

INTERNAL="YES|no"

Optional.

Default: INTERNAL="**YES**"

Refresh: Yes

Specifies whether the HCI is to support TPs attached as subtasks within the HCI address space. Compuware recommends specifying **YES** or accepting the default for all current implementations of the HCI.

JESCHAR="$".

Optional.

Default: JESCHAR="**$**" (U.S. dollar sign)

Refresh: Yes

Specifies the command character used from the master console or from an SDSF log screen to issue commands to the JES2 or JES3 subsystem. The value of this attribute is used by the HCI as the first character of the command used to cancel TP programs that have been started by the Server Activation Facility, but which have not registered with the HCI within the allowable time limit.

For example, the command for JES2 using the default character:

```
$CJnnnnn
```

The same command for JES2 using the forward slash (/):

```
/CJnnnnn
```

In this way, multiple JES subsystems can support HCI TP programs.

JESID="2|3"

Optional.

Default: JESID="**2**"

Refresh: No

Specifies that either JES2 or JES3 is the primary subsystem for the LPAR on which the HCI is executing. Various functions within the HCI depend on the correct setting of this attribute.

JESJPRM="NO|yes"

Optional.

Default: JESJPRM="**NO**"

Refresh: Yes

Specifies whether the HCI should add a /*JOBPARM SYSAFF=* for JES2 systems or a //*MAIN SYSTEM=xxxxxxxx for JES3 systems to JCL submitted by the HCI to start TP programs. If your JCL for TP programs is to run on systems other than the one on which the HCI is executing, then omit this attribute or specify JESJPRM="**NO**".

JRNMASK="0000000000000000|FFFFFFFFFFFFFFFF"

Optional.

Default: JRNMASK="**0000000000000000**"

Refresh: Yes

Specifies exactly 16 hexadecimal digits (0-9 and A-F) that are used to control which journal events are to be collected when journaling is active.

"**0000000000000000**"

Causes no journal events to be written to the HCI journal.

"**FFFFFFFFFFFFFFFF**"

Causes all journal events to be written to the HCI journal.

MAININT="nnnnnn"

Optional.

Default: MAININT="**1500**"

Refresh: Yes

Specifies the interval for the main timer in the HCI. This number is specified in hundredths of a second. Valid values are 1 through 360000. The default (15 seconds) is suitable for all installations. If this value is increased appreciably, the time between the termination of a TCP/IP connection or the termination of a TP program and when the HCI processes this event may be increased.

MAXCCBS=""nnnnn"

Optional.

Default: MAXCCBS="**1024**"

Refresh: Yes

Specifies the maximum number of *concurrent* conversations or connections that the HCI will support. Setting this attribute to a large number increases private storage usage, but otherwise does not cause any negative effects. Valid values are 128 through 32768. Each CCB is 1376 (decimal) bytes long.

MAXDCBS="nnnnn"

Optional.

Default: MAXDCBS="**32**"

Refresh: Yes

Specifies the maximum number of *concurrent* discrete destinations that the HCI will support. Setting this attribute to a large number increases private storage usage, but otherwise does not cause any negative effects. Valid values are 8 through 32768. Each DCB is 760 (decimal) bytes long.

MAXJRES="nnnnn"

Optional.

Default: MAXJRES="**8192**"

Refresh: Yes

Specifies the maximum number of *concurrent* journal requests that the HCI will support. Setting this attribute to a large number increases private storage usage, but otherwise does not cause any negative effects. In addition, a large number ensures that no journal records are lost due to JRE exhaustion. Valid values are 512 through 32768. Each DCB is 16 (decimal) bytes long.

MAXLUBS="nnnnn"

Optional.

Default: MAXLUBS=”**16**”

Refresh: Yes

Specifies the maximum number of SNA LU2 sessions that the HCI will support. If you are not supporting any LU2 remote partners, then allowing this attribute to use the default is sufficient. Valid values are 1through 2768. Each LUB is 496 (decimal) bytes long.

MAXRCBS="nnnnn"

Optional.

Default: MAXRCBS=”**32**”

Refresh: Yes

Specifies the maximum number of concurrent TP program address spaces that the HCI will support. Each address space can be associated with multiple TP users, but in most cases it is a one-for-one correspondence. Region control blocks (RCBs) are allocated from ECSA (SP=241), and therefore this number should not be excessively bigger than necessary. Valid values are 8 through 32768. Each RCB is 96 (decimal) bytes long.

MAXUIBS="nnnnn"

Optional.

Default: MAXUIBS=”**32**”

Refresh: Yes

Specifies the maximum number of concurrent TP program TPNAMES that the HCI will support. User interface blocks (UIBs) are allocated from ECSA (SP=241), and therefore this number should not be excessively bigger than necessary. Valid values are 8 through 32768. Each UIB is 1040 (decimal) bytes long.

MAXWRES="nnnnn"

Optional.

Default: MAXWRES=”**4096**”

Refresh: Yes

Specifies the maximum number of concurrent work requests that the HCI will support. Work request elements (WREs) are allocated from ECSA (SP=241), and therefore this number should not be excessively bigger than you find necessary. Valid values are 8 through 32768. Each WRE is 1040 (decimal) bytes long.

OACBINT="nnnnn"

Optional.

Default: OACBINT=”**3000**”

Refresh: Yes

Specifies the time in hundredths of a second that the HCI will allow to elapse between attempts to open a VTAM ACB specified by the ACBNAME attribute of an HCICNAMB element. Valid values are 1 through 360000.

OPCMD="WTOR|modify"

Optional.

Default: OPCMD=”**WTOR**”

Refresh: Yes

Specifies how operator commands are entered to the HCI.

"WTOR"

Specifies that a Reply to Operator command (**r nn**,) is to be used.

"MODIFY"

Specifies that the z/OS MODIFY (**F**) command is to be used.

PACING="nnnnn"

Obsolete.

Default: PACING="**0**"

Refresh: N/A

No longer used by the HCI. If the attribute is specified, the value must be **0**.

PREPROC="xxxxxxxx"

Optional.

Default: PREPROC="**HCIYPREP**"

Refresh: Yes

Specifies the name of a procedure in SYS1.PROCLIB (or an appropriate user procedure library) that starts the initialization of the secondary HCI on each member of a parallel sysplex. The main HCI is initialized by the customer via JCL or a started task. The HCI automatically routes z/OS **START** commands to each other member of the sysplex indicating that the PROC named in the PREPROC attribute be started. Name this PROC any maximum eight-character name, but the PROC must be available to every member of the sysplex. See "Sysplex Routed Tasks" on page 4-2 for more information about this initialization procedure.

If the SYSPLEX attribute is omitted (indicating no sysplex support requested), then the PREPROC attribute is ignored.

ROUTCMD="NO|yes"

Optional.

Default: ROUTCMD="**NO**"

Refresh: Yes

Specifies whether TP programs that are initiated by START commands are to be routed to the member of the sysplex indicated on the member in the HCI //PARMLIB DD.

"NO"

Specifies that the z/OS START command is to be issued only on the system on which the primary HCI is running.

"YES"

Specifies to route z/OS START commands to each z/OS in the SYSPLEX.

SECFAC="xxxx...xxxx"

Optional.

Default: None

Refresh: Yes

Specifies the name of a Facility Class entity that is to be checked when a TP program issues a CWOPER call in order to issue an operator command. The user ID under which the TP is executing must have ACCESS(UPDATE) to the Facility Class entity in order to issue the CWOPER call to issue the REFRESH command. No other operator commands may be issued from this call and none is protected by the RACF entity.

SRVRJOB="xxxxx"

Optional.

Default: SRVRJOB=*"***HCITP***"*

Refresh: Yes

Specifies a one- to five-character job name prefix that the HCI uses for all TP programs submitted by the HCI. A one- to three-digit numeric suffix is added onto this prefix and incremented each time the job is submitted in order that the job names will be unique. When the suffix reaches 9, 99, or 999 (depending upon the length of the prefix), the number wraps to 0, 00, or 000.

SYSPLEX="xxxx"

Optional.

Default: None

Refresh: No

Specifies a four-character subsystem ID to be assigned to each of the sysplex HCI members. The value must be exactly four characters in length and must consist of uppercase letters and numbers only.

Omitting this attribute specifies that you do *not* want sysplex support to be activated.

TCPAMSG="NO|yes"

Optional for TCPACCESS only.

Default: TCPAMSG=*"***NO***"*

Refresh: Yes

Specifies whether the TCPACCESS TERROR macro is to be used to display error messages if your TCP/IP protocol stack is CA's TCPACCESS product.

*"***NO***"*

Sends TCPACCESS messages to the //HCIPRINT DD.

*"***YES***"*

Use the TCPACCESS TERROR macro to display error messages.

TPSEC="NO|yes"

Optional.

Default: TPSEC=*"***NO***"*

Refresh: Yes

Specifies whether you want the HCI to ensure that the user ID under which a TP program is executing has access to the TPNAME that it is connecting to.

*"***NO***"*

Specifies that no RACF checking of the user ID versus the TPNAME is to occur.

*"***YES***"*

Specifies that the user ID must have ACCESS(READ) to the FACILITY CLASS entity in order for the HCI to allow execution of the TP. The FACILITY CLASS entity name is constructed by prefixing the TPNAME with the four-character subsystem ID of the HCI.

/>

Close HCICNGCA element. You must include a forward slash and right angle bracket (/>) XML construct to close the <HCICNGCA element. Failure to do so causes the document to not be well-formed and parsing to fail.

<HCICNAMB Element

Include at least one <HCICNAMB element in the XML document, but any number can be specified. Each one specifies an ACB name for LU6.2 and LU2 communication to be opened and used for inbound and outbound connections. For TCP/IP, the <HCICNAMB element specifies the name of a TCP/IP region active on this LPAR. The LUTYPE= attribute specifies what kind of connection (VTAM or TCP/IP) is to be created by this HCICNAMB element.

For the refresh capability, existing AMB's can be updated via an HCICNAMB element in the refresh document and new AMB's can be added. The ACBNAME or TCPNAME is used to associate an AMB element in the refresh document with ones already defined by the initial document. For existing AMB's only the TCPPSWD can be changed dynamically. New AMB's can contain any of the defined attributes. The <HCICNAMB element attributes are described below.

LUTYPE="lutype"

Required.

Default: None

Refresh: No

Specifies one of the following valid types indicating the access method and protocol to be used for all connections/conversation defined by this HCICNAMB element:

- TCPIP
- LU62
- LU2
- TCPACCESS.

RCVLN="nnnnnnnn"

Required.

Default: None

Refresh: No

Specifies the maximum data length in bytes that can be received on the connection/conversation described by the current HCICNAMB element. Valid values are 4096 through 16777211.

TCPNAME="xxxxxxxx"

Optional. Default: [*Default TCP/IP region name*]

Refresh: No

Specifies the name of a TCP/IP region that is to process connections to/from this HCICNAMB element. If you have multiple active TCP/IP regions and you want a particular one to be chosen, then specify the region name here. From any ISPF command line, you can issue the **TSO NETSTAT HOME** command to find the name of your TCP/IP region.

ACBNAME="xxxxxxxx"

Required if LUTYPE is LU62 or LU2.

Default: None

Refresh: No

Specifies the VTAM ACBNAME attribute that is to be used by conversations associated with this HCICNAMB.

GENNAME="xxxxxxxx"

Optional.

Default: None

Refresh: No

Specifies the VTAM generic name to be used by conversations associated with this HCICNAMB. The generic name is moved into the node initialization block (NIB) prior to use. If this name is present, the HCI indicates to VTAM that it is to be known by this generic name, and a SETLOGON is issued to add the generic name. Multiple HCICNAMB elements may specify the same generic name, and multiple HCIs on the sysplex may also specify the same name.

HPNS="CURR"

Optional.

Default: HPNS="**CURR**"

Refresh: No

Specifies the only valid value for the HPNS macro attribute for HCI Release 3.0. In prior HCI releases, the HPNS attribute caused a particular TCP/IP interface to be used.

IPV6="NO|yes"

Optional.

Default: IPV6="**NO**"

Refresh: No

Specifies whether the HCI should attempt to use the TCP/IP IPv6 interface for connections to/from remote hosts and this HCI. See “IPv6 Support” on page 1-2 for a more complete description.

If your TCP/IP region on this LPAR does *not* support IPv6, then specify IPV6="**NO**"; otherwise, an error message is generated when an attempt is made to obtain an IPv6 socket. If the IPv6 SOCKET obtain fails, the HCI reverts to an IPv4 socket and communication proceeds as usual. However, if the TCP/IP region on this LPAR does support IPv6, specify IPV6="**YES**" in order to take advantage of this support.

Note that specifying **“YES”** does not necessarily mean that communication with the remote host will be via IPv6. If the remote host is connected by an IPv4 network, or if the host itself does not support IPv6, then IPv4 is used.

MODEL="n"

Optional.

Default: None

Refresh: No

Applies only to HCICNAMB elements specifying LUTYPE=LU2. This attribute specifies whether the 3270 model supported by the HCI is a model 2, 3, 4, or 5. Specifying the correct model ensures that an appropriate BIND image can be sent to the remote partner LU. Omit this attribute if the partner LU2 model is not known.

PGMNAME="xxxxxxxx"

Optional and ignored by HCI Release 3.0. Default: Program name appropriate for the LUTYPE specified.

Specifies the main program name to be invoked for the LUTYPE specified on the HCICNAMB element. Omit this attribute and allow the HCI to supply the correct name. You may see this name in the XML generated by the HCI and which is written to the //HCIPRINT DD. You may leave this attribute in the XML or delete it.

TCPPSWD="xxxxxxxx"

Optional.

Default: None

Refresh: Yes

Applies an optional password to TCPACCESS connections. Omit this attribute for any but TCPACCESS HCICNAMB elements. Supply it only if a password has been specified in the TCPACCESS definition datasets.

/>

Close HCICNAMB element. You must include a forward slash and right angle bracket (/>) XML construct to close the <HCICNAMB element. Failure to do so causes the document to not be well-formed and parsing to fail.

<HCICNPCB Element

At least one <HCICNPCB element must be present in the XML document if you are defining any TCP/IP connections. Any number can be specified.

For the refresh capability, existing PCBs can be updated or new ones can be added. Port number is used to define an existing PCB. When a new PCB is added all the attributes below can be specified. The Refresh specification applies to updating existing PCBs only. The <HCICNPCB element attributes are described below.

PORT="nnnnn"

Required.

Default: None.

Refresh: No

Specifies the port number associated with the connection described by this HCICNPCB element. Valid port numbers are from 1 through 65535. Check with your network administrator to ensure that you have not accidentally specified a port number already in use.

The port number is used as follows:

- For "TYPE=**LOCAL**" (inbound connections), the PORT number specifies the port on which the HCI *listens* for incoming connection requests.
- For "TYPE=**REMOTE**" (outbound connections), the PORT number specifies the port on the remote host to which connections are to be made.
- For "TYPE=**BOTH**" (both inbound and outbound), the PORT number specifies both the inbound *listen* port and the outbound *connection* port.

TCPNAME="xxxxxxxx"

Optional.

Default: [*Default TCP/IP region name*]

Refresh: No.

Specifies the name of a TCP/IP region that is to process connections to/from this HCICNPCB element. If you have multiple active TCP/IP regions and you want a particular one to be chosen, then specify the region name here. This name must match a TCPNAME attribute in an <HCICNAMB element.

TPNAME="xxxx...xxxx"

Required.

Default: None

Refresh: Yes

Specifies the TP program name that is also specified in an HCICNTPT element associated with connections defined by this HCICNPCB element. Valid TP program names are 1 through 64 characters in length.

DATAENC="ASCII|EBCDIC"

Optional.

Default: DATAENC=**"ASCII"**

Refresh: No.

Specifies the data encoding used by TP programs for connections defined by this HCICNPCB element. This attribute is not used by HCI Release 3.0, but if specified must be one of the values shown.

DVIPA="IPv4|IPv6 address"

Optional.

Default: None

Refresh: Yes

Specifies the address (IPv4 or IPv6) from which connections are accepted for this HCICNPCB element if TCP/IP VIPARANGE processing has been defined in the TCP/IP configuration datasets. This address overrides the IP address of the host on which the HCI is running, thus allowing you to move the HCI from one LPAR to another retaining the IP address known to the network clients.

The address must be in presentation form for one of the following:

- **IPv4 addresses**: nnn.nnn.nnn.nnn, where *nnn* is a decimal number from 0 through 255 for each of the four sub-elements.
- **IPv6 address:** hhhh:hhhh::hhhh where *hhhh* is a hexadecimal number (0-9, A-F) for each of the one to eight sub-elements.

DVIPN="IP name"

Optional.

Default: None

Refresh: Yes

Specifies the 1- to 64-character name, known to your HOSTS file or to your DNS server, from which connections are accepted for this HCICNPCB element If TCP/IP VIPARANGE processing has been defined in the TCP/IP configuration datasets. This name overrides the IP name of the host on which the HCI is running thus allowing you to move the HCI from one LPAR to another retaining the IP name known to the network clients.

RECFM="F|U|V"

Required.

Default: None

Refresh: Yes

Specifies the format of records received and sent by the HCI for connections defined by this HCICNPCB element. The valid values are

- **"F":** Fixed length. Length specified by the LRECL attribute
- "**U**": Undefined length. Length specified by the EORCHRS and EOBCHRS attributes
- "**V**": Variable length. Length specified by the LLZZ attribute

EORCHRS="hhhh...hhhh"

Optional.

Default: None

Refresh: Yes

Specifies the hexadecimal characters that define the end of a record if RECFM=”**U**” for this HCICNPCB element. When the HCI recognizes this set of characters, it considers the record complete and passes it to the TP program in response to a READ call. Valid values are 2 through 16 hexadecimal characters.

If there is a differentiation between End-of-Record and End-of-Block, then also specify the EOBCHRS attribute.

EOBCHRS="hhhh...hhhh"

Optional.

Default: None

Refresh: Yes

Specifies the hexadecimal characters that define the end of a block of data if RECFM=”**U**” for this HCICNPCB. When the HCI recognizes this set of characters, it considers the block of input data complete and passes it to the TP program in response to a READ call. Valid values are 2 through 16 hexadecimal characters.

LRECL="nnnnn"

Optional.

Default: LRECL=”**4096**”

Refresh: Yes

Specifies the size of each logical record to be passed to the TP program if RECFM=”F” for this HCICNPCB. Valid values are 16 through 32768 bytes.

LLZZ="(s1,s2,s3,s4,s5,s6)"

Optional.

Default: None

Refresh: Yes

Specifies the location, length, type, and disposition of the length field that exists on each block of data if RECFM="**V**" for this HCICNPCB. The six subfields must all be present in the following order

1. Starting location in the record where the length field begins. Valid values are 000 through 32768.
2. Length of the length field. Valid values are 1 through 4.
3. Encoding of the length field: **CHARA** (ASCII character), **CHARE** (EBCDIC character), **BINB** (binary big endian), or **BINL** (binary little endian).
4. **INCL** if the length includes itself, or **EXCL** if the length does not include itself. Note that *INCL* is the way variable length records on DASD are defined.
5. **MAKE** if the HCI is to provide the length field on output, or **USE** if the length field already exists in the output message.
6. **DELETE** if the HCI is to remove the length field from inbound data, or **KEEP** if the length field is to be passed to the TP program on input messages.

PROTOCOL="UNIFACE|UNIFACE8|GCS"

Optional.

Default: None

Refresh: Yes

Specifies one of the three Compuware product protocols. If connections defined by this HCICNPCB element are to be processed by one of the Compuware product protocols, then omit the other following formatting attributes, whose values will automatically be set by the HCI:

- DATAENC
- LLZZ
- RECFM
- LRECL.

TYPE="LOCAL|REMOTE|BOTH"

Optional.

Default: TYPE=”**LOCAL**”

Refresh: No.

Specifies whether this HCICNPCB element represents a *listening* port (TYPE="**LOCAL**") or a *connecting* port (TYPE=”**REMOTE**") or both listening and connecting. Further, TYPE="**LOCAL**" implies that the port number in this HCICNPCB is a port on this host. TYPE="**REMOTE**" implies that the port number in this HCICNPCB is a port on the remote host.

/>

Close HCICNPCB element. You must include a forward slash and right angle bracket (/>) XML construct to close the <HCICNPCB element. Failure to do so causes the document to not be well-formed and parsing to fail.

<HCICNJCB Element

One <HCICNJCB element for each journal dataset you have defined, initialized, and intend to use must be present in the XML document. See Chapter 6, “Using the Journaling Facility” for more information about the HCI’s journaling facility. Compuware recommends that you define three journal datasets of sufficient size each to hold ten minutes of activity.

Even though you have specified these datasets in <HCICNJCB elements, nothing is written to them (and no overhead is encountered) if the journal mask attribute (**JRNMASK=)** in the <HCICNGCA element is set to all zeros. You can change this setting dynamically by operator commands if you want to start and stop journal activity. However, if no journals are defined, you cannot dynamically start capturing journal records.

The <HCICNJCB element attributes are described below.

DDNAME="xxxxxxxx"

Required.

Default: None

Refresh: Yes

Assigns a one- to eight-character DD name to the journal whose dataset name is specified by the DSNAME attribute.

DSNAME="xxxxx...xxxxx"

Required.

Default: None

Refresh: Yes

Assigns a 1- to 35-character dataset name to the journal whose DDNAME is specified by the DDNAME attribute. The name must have been initialized as documented in Chapter 6, “Using the Journaling Facility”.

/>

Close <HCICNJCB element. You must include a forward slash and right angle bracket (/>) XML construct to close the <HCICNJCB element. Failure to do so causes the document to not be well-formed and parsing to fail.

<HCICNSIT Element

One <HCICNSIT element for each *outbound* connection that a TP program wants to make must be present in the XML document. Although originally designed for LU 6.2 conversations, the side information table (SIT) concept and implementation are applicable to TCP/IP as well. Each <HCICNSIT element defines the parameters for an outgoing connection/conversation in such a way that the TP program itself need not be aware of many of these values.

The refresh facility can update existing SIT elements and can add new ones. The SYMNAME attribute is used to define an existing or new SIT. For new SITs all attributes can be specified in the refresh document. For existing SITs, the Refresh specification on each attribute below defines whether it can be changed or not. The <HCICNSIT element attributes are described below.

SYMNAME="xxxxxxxx"

Required.

Default: None

Refresh: No

Assigns a symbolic destination name for TP programs to specify for connections defined by this HCICNSIT element. This symbolic name must be known to the TP program. Therefore, you must coordinate assigning a name with the development group. Compuware products that use the HCI will define the names you are to use. See the individual Compuware product's installation guide for requirements for the HCICNSIT elements.

The valid value for this attribute is a one- to eight-character name known to the TP program for its outbound connection/conversation requests.

LUTYPE="lutype"

Required.

Default: None

Refresh: No.

Specifies one of the following valid *lutypes* indicating the access method and protocol to be used for all connections/conversation defined by this <HCICNSIT element:

- TCPIP
- LU62
- TCPACCESS.

BYPDLAY="NO|yes"

Optional for TCP/IP only.

Default: BYPDLAY="**NO**"

Refresh: Yes

Specifies whether upon TCP/IP close the system-defined delay interval is to be invoked in order to ensure that all TCP/IP traffic has exited the network and has been processed by the remote host.

"**NO**”

Specifies that this delay is to be used, which causes a noticeable performance implication of waiting for this period of time.

"YES"

Specifies that the no delay period is to be waited and that the connection is to be closed immediately.

IPADDR="nnn.nnn.nnn.nnn"|"hhhh:...hhhh"

Optional for TCP/IP only.

Default: None

Refresh: Yes

Specifies the IPv4 or IPv6 address of the remote host with which connections defined by this HCICNSIT entry are to be made. If you specify this attribute, then omit the IPNAME attribute. The maximum length of the IP address value is 15 bytes for IPv4; and, 39 bytes for IPv6. The valid presentation form of the IPv4 address is dotted decimal (10.10.10.10). The valid form for IPv6 is colon hexadecimal (0101:01FF:AAFF:CE01::)

IPNAME="xxxxx...xxxxx"

Optional for TCP/IP only.

Default: None

Refresh: Yes

Specifies the name, known to your DNS or the local HOSTS dataset, of the remote host with which connections defined by this HCICNSIT entry are to be made. If you specify this attribute, then omit the IPADDR attribute. The maximum length of this IPNAME value is 64 bytes. If the name is longer than this maximum, then you must use IPADDR instead.

PORT="nnnnn"

Required for TCP/IP only.

Default: None

Refresh: Yes

Specifies the port number upon which the remote partner host is listening and to which connections defined by this HCICNSIT element are to be made. Valid port numbers are from 1 through 65535.

TCPNAME="xxxxxxxx"

Optional for TCP/IP only.

Default: None

Refresh: Yes

Specifies the name of a TCP/IP region running on this LPAR through which connections to the remote partner host defined by this HCICNSIT element are to be made. If you specify this attribute, then you must also specify an <HCICNAMB element with the same TCPNAME. Hence the connection to the desired protocol stack can be made. If this attribute is omitted and if TCPNAME is omitted from all <HCICNAMB elements, then the first one defined is used for these outbound connections.

LLUNAME="xxxxxxxx"

Optional for LU6.2 only.

Default: LLUNAME=[*First HCICNAMB ACBNAME*]

Refresh: Yes

Specifies the ACBNAME of one of the <HCICNAMB element entries in order that a particular one be used as the local LU for conversations defined by the <HCICNSIT element. If this attribute is omitted for LU 6.2 conversations, then the first LU 6.2 <HCICNAMB element is used as a default.

MAXS="0"

Optional for LU6.2 only.

Default: MAXS=*"***0***"*

Refresh: Yes

Has been removed from use and VTAM now supplies this information automatically for LU 6.2 conversations. . If you specify this attribute, you must specify its value as "**0**".

MINCW="0"

Optional for LU6.2 only.

Default: MINCW=*"***0***"*

Refresh: Yes

Has been removed from use and VTAM now supplies this information automatically for LU 6.2 conversations. . If you specify this attribute, you must specify its value as "**0**".

MINCL="0"

Optional for LU6.2 only.

Default:MINCL=*"***0***"*

Refresh: Yes

Has been removed from use and VTAM now supplies this information automatically for LU 6.2 conversations. If you specify this attribute, you must specify its value as "**0**".

MODNAME="xxxxxxxx"

Optional for LU6.2 only.

Default: None

Refresh: Yes

Specifies the Logon Mode Table entry name to be used for the parameters for the outbound conversation defined by this <HCICNSIT element.

PARSESS="YES|NO"

Optional for LU6.2 only.

Default: PARSESS=*"***YES***"*

Refresh: Yes

*"***YES***"*

Specifies that the LU associated with this outbound conversation supports parallel sessions and these should be initiated when the session is established.

*"***NO***"*

Specifies that the LU associated with this outbound conversation does not supports parallel sessions.

PASSWD="******"**

Optional for LU6.2 only.

Default: None

Refresh: Yes

Assigns a one- to eight-character password for the user ID specified in this HCICNSIT element. The user ID/password combination is sent to the partner LU in the FMH-5 and is used by the partner to ensure that the initiator of the conversation is valid. If the PASSWD attribute is specified, the USERID attribute must be specified as well. Note that PASSWD is not displayed in the //HCIPRINT DD dataset.

PLUNAME="xxxx...xxxx"

Optional for LU6.2 only.

Default: None

Refresh: Yes

Specifies the 1- to 17-character partner LU name for the remote end of the conversations established by this HCICNSIT entry. Specify the value as NETID.LUNAME, for a maximum of 17 bytes in length. This value can be supplied by the local TP program before allocating a conversation with the remote partner.

TPNAME="xxxxx...xxxx"

Optional for LU6.2 only.

Default: None

Refresh: No.

Specifies the 1- to 64-character name of the TP program running in the partner LU as defined by PLUNAME attribute.

USERID="******"**

Optional for LU6.2 only.

Default: None

Refresh: Yes

Assigns a 1 - 8 character user ID that is sent to the partner LU in the FMH-5 and is used by the partner to ensure that the initiator of the conversation is valid.

/>

Close <HCICNSIT element. You must include a forward slash and right angle bracket (/>) XML construct to close the <HCICNSIT element. Failure to do so causes the document to not be well-formed and parsing to fail.

<HCICNTPT Element

One <HCICNTPT element for each *inbound* connection that a TP program wants to make must be present in the XML document. Although originally designed for LU 6.2 conversations, the teleprocessing program table (TPT) concept and implementation are applicable to TCP/IP as well. Each <HCICNTPT element defines the parameters for an inbound connection/conversation in such a way that the TP program itself need not be aware of many of these values.

The <HCICNTPT element attributes are described below.

TPNAME="xxxx...xxxx"

Required for LU6.2 and TCP/IP.

Default: None

Refresh: Yes

Specifies the name the TP program is known by. The program itself must know this name from its own sources, and the TP identifies itself to the HCI in the CWRNU call by this name. Valid TP program names are 1 through 64 characters in length.

A required <HCICNPCB element in this XML document must specify the same TPNAME attribute value as is specified here. These two names must match exactly in case and length.

ALIASOF="xxxx...xxxx"

Optional for LU6.2 and TCP/IP.

Default: None

Refresh: Yes

Specifies a TP program name that acts as the primary name for this alias. The ALIASOF attribute value must specify a TPNAME that is defined in its own <HCICNTPT element within this XML document. A TP program can register with the HCI as either the TPNAME specified in this element or the TPNAME specified in the ALIASOF attribute.

MEMBER="xxxxxxxx"

Optional for LU6.2 and TCP/IP.

Default: None

Refresh: Yes

Specifies the name of a member in the partitioned dataset (PDS) defined in the HCI execution JCL with DDNAME //HCIPARM DD. This member contains instructions to the HCI as to how the TP program defined by this HCICNTPT element is to be started. That is, the member contains JCL, a START command, an ATTACH command, or a ROUTE command. See Chapter 4, "Using the Server Activation Facility" for a complete discussion of this subject.

MAXCNV="nnnnn"

Optional for LU6.2 and TCP/IP.

Default: MAXCNV="**1**"

Refresh: Yes

Specifies the maximum number of conversations that one instance of the TP program can handle and applies only to TP programs capable of managing multiple conversations or connections simultaneously. The minimum number is 1, and the maximum number is not defined. When one instance of this TP program reaches its maximum, the next inbound connection or conversation request causes a new instance of the TP program to be started. New conversations/connections are passed to an existing instance of this TP program in a *round robin* manner until all instances currently executing are at this maximum value.

If you specify a value greater than 1, also specify the MULTSEC="**YES**" attribute.

For single-threaded TP programs that cannot handle multiple conversations, specify MAXCONV="**1**" or allow this attribute to default. Hence each conversation or connection request results in a new instance of the TP program being started.

MAXTPS="nnnnn"

Optional for LU6.2 and TCP/IP.

Default: MAXTPS="**999**"

Refresh: Yes

Specifies the maximum number of instances of this TP program which will be initiated by the HCI and applies to all TP programs whether managing multiple conversations. When this maximum is reached, subsequent conversation or

connection requests are denied. The minimum number is 1, and the maximum number is not defined.

MULTSEC="NO|yes"

Optional for LU6.2 and TCP/IP.

Default: MULTSEC=**"NO"**

Refresh: Yes

Specifies whether the TP program defined by this HCICNTPT element is capable of handling multiple simultaneous conversations or connections.

"NO"

Specifies that only one conversation or connection can be processed at one time by this TP program.

"YES"

Specifies that this TP program can handle multiple simultaneous conversations or connections. If this attribute is set to MULTSEC="YES", specify a value greater than 1 for the MAXCNV attribute to indicate how many conversations or connections the TP program can handle at one time

MAXPTIM="nnnnn"

Optional for LU6.2 and TCP/IP.

Default: MAXPTIM=**"1440"**

Refresh: Yes

Specifies the number of seconds that the HCI will wait after a TP program starts before a conversation or connection request arrives from a remote host or LU. If this time is exceeded, the connection or conversation request is denied and the TP program is canceled. Valid values are 0 through 32768 seconds.

MAXSTIM="nnnnn"

Optional for LU6.2 and TCP/IP.

Default: MAXSTIM=**"1440"**

Refresh: Yes

Specifies the number of seconds that the HCI will wait after submitting or starting a TP program before that program issues a Register New User call indicating that it is ready to handle the inbound connection or conversation request. If this time is exceeded, the connection or conversation request is denied and the TP program is canceled. Valid values are 0 through 32768 seconds.

RECVLEN="nnnnn"

Optional for LU6.2 and TCP/IP.

Default: RECVLEN=**"4096"**

Refresh: Yes

Specifies the maximum length in bytes of any message to or from the remote host or TP connected to the TP program defined by this HCICNTPT element. Valid values are 4096 through 16777211.

SRVRJOB="xxxxxxxx"

Optional for LU6.2 and TCP/IP.

Default: None

Refresh: Yes

Specifies a job name prefix to be used for TP programs submitted by the HCI defined by this HCICNTPT element. This attribute overrides the SRVRJOB attribute for this TP

program only. If this attribute is omitted, then the SRVRJOB attribute is used as the job name prefix.

USERSEC="YES|no"

Optional for LU6.2 and TCP/IP.

Default: USERSEC=**"YES"**

Refresh: Yes

Specifies whether RACF (ACF/2 or TOPSECRET) RACROUTE calls are to be made to ensure that the user ID under which a TP program is executing has the authority to assign itself as the TP name specified in this HCICNTPT element and in the associated register new user call.

"YES"

Specifies that the facility class entity named the same as the TP name is to be checked and the user ID under which this TP program is executing must have at least ACCESS(READ) to this entity. If it does not have this access, then the register new user call fails. If USERSEC="YES" is specified, but no facility class entity named by the TPNAME exists, then access is allowed.

"NO"

Specifies that RACF (ACF/2 or TOPSECRET) RACROUTE calls are not to be made.

DSNAME="xxxx.xxxx"

Ignored for all connection types.

Default: None.

Refresh: N/A

The DSNAME attribute has been removed from the HCI and is no longer processed. The MEMBER attribute can only specify the name of a member in the //HCIPARM DD dataset, and no optional dataset is supported.

CATEGORY="xxxx...xxxx"

Optional for LU6.2 and TCP/IP.

Default: None

Refresh: Yes

Specifies a 1- to 16-byte name used by the TP program to associate different HCICNTPT elements with a particular name from a set of them. This attribute is not used by the HCI, but is passed to a TP program issuing an extract TP program list extension call.

WILDCRD="NO|yes"

Ignored for all connection types.

Default: WILDCRD=**"NO"**

Refresh: Yes

Specifies whether this HCICNTPT element represents a generic name for a set of TPs.

"NO"

Specifies not to use this HCICNTPT element's value for the TPNAME attribute as a generic name.

"YES"

Specifies to use this HCICNTPT element's value for the TPNAME attribute as a generic name. That is, the TPNAME becomes a prefix for all the real TP names that are used by the TPs themselves to register. This capability allows TPs to start

by themselves, to be referred to by the generic name, and to each process a single conversation at a time.

/>

Close <HCICNTPT element. You must include a forward slash and right angle bracket (/>) XML construct to close the <HCICNTPT element. Failure to do so causes the document to not be well-formed and parsing to fail.

Chapter 10.
Using HCI-XPERT

HCI-XPERT is a collection of two diagnostic tools and three utilities. One diagnostic tool, HCILOOK, provides analysis and changing of pertinent run-time controls while the other, HCITELL, provides analysis of journal data gathered for problem determination purposes. The IMPORT and EXPORT utilities, provide a method to package the journal data for its transmission (via FTP) to Compuware should the need arise to resolve a problem. The MESSAGES utility enables the viewing of some HCI abend codes and workstation error codes.

HCI-XPERT executes under TSO ISPF and supports its command structure, and all screens support the ISPF HELP command (usually *PF1*). The help tutorial screens explain all pertinent information about the screen including commands available and field definitions.

This discussion of HCI-XPERT is not all-inclusive; consult the tutorial panels provided. More detailed information is provided for getting started than for the remaining part of the facility. The panels are shown for a 24x80 terminal size. Currently all 3270 models 2, 3, 4, and 5 are supported.

Access Considerations

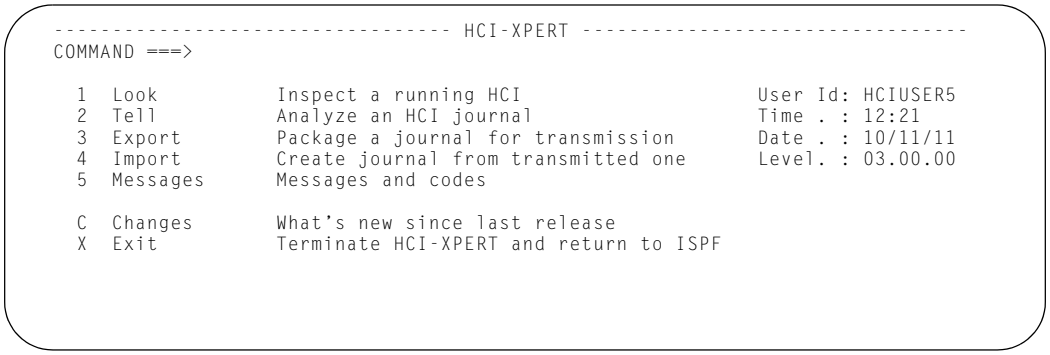
HCI–XPERT consists of a CLIST, two REXX EXECs, several assembler routines, many ISPF screens and tutorial definitions. Several libraries must be accessible in order for HCI–XPERT to function. Access to these libraries may be provided via a system library or may need to be defined by the user as follows.

- **//SYSEXEC DD**: Ensure that your TSO LOGON allocates the DDNAME SYSEXEC to a PDS that contains HCI–XPERT's REXX EXECs.
- **//ISPPLIB DD**: Ensure that your TSO LOGON allocates the DDNAME ISPPLIB to a PDS that contains the HCI ISPF panels.
- **//ISPLLIB DD**: Ensure that your TSO LOGON allocates the DDNAME ISPLLIB to a PDS that contains the HCI load modules.
- **//ISPTLIB DD:** Ensure that your TSO LOGON allocates the DDNAME ISPTLIB to a PDS that contains the HCI tables (HCIJCMDS module).

INVOKING HCI-XPERT

You can invoke HCI-XPERT by entering the command **TSO HCIXPERT** on the ISPF COMMAND line. Figure 10-1 on page 10-2 shows the HCI-XPERT main menu.

Figure 10-1. HCI-XPERT Main Menu



Because the HCITELL and IMPORT facilities are intended primarily for use by Compuware Customer Support for HCI debugging and diagnosis, documentation for these tools is not included in this user/reference guide.

HCILOOK Facility

The HCILOOK facility lets you examine the current status of the HCI dynamically to analyze on-going events, and to perform certain update functions to change the configuration defaults that were set in the configuration assembly. Additionally, you can use HCILOOK to unregister a TP and/or dynamically (and forcibly) cancel it. HCILOOK provides information about the following:

- Pertinent HCI tuning data
- Active HCI regions
- Registered TPs
- Active sessions
- Executed module data.

Invoking HCILOOK

Upon entering **1** (Look) on the command line of the HCI-XPERT main menu (Figure 10-1), you are asked to enter the four-character subsystem ID. To enable the update capability, enter the subsystem ID followed by a space and the word *UPDATE* (xxxx UPDATE).

Running HCILOOK without update activated is slightly more efficient than running it with the update capability active.

HCI Subsystem Display

The HCI SUBSYSTEM DISPLAY screen, shown in Figure 10-2 on page 10-3) lists all HCI subsystems known to z/OS. Instead of entering the four-character subsystem ID when you are prompted, substitute instead the single character question mark (?). Doing so displays a list of all z/OS subsystems that are an HCI subsystem.

Figure 10-2. HCI SUBSYSTEM DISPLAY Screen

```
--------------------------- HCI SUBSYSTEM DISPLAY ---------- Row 1 to 17 of 41
COMMAND   ===>                                               SCROLL ===> PAGE

  ENTER -D- TO DISPLAY HCI SUBSYSTEM

    ADDRESS   NAME  SSCTSUSE  SSCTSUS2  STATUS

_   00B79AD0  HCID  80C8C3C9  2F42A000  ACTIVE
_   00B795A0  MHCI  80C8C3C9  2F42B000  ACTIVE
_   00ED4AD0  Q500  80C8C3C9  2F41C000  ACTIVE
_   00ED45A0  Q520  80C8C3C9  2F41E000  ACTIVE
_   00ED4070  Q510  80C8C3C9  2F41B000  ACTIVE
_   00ED0AD0  PHCI  80C8C3C9  2F60A000  ACTIVE
_   00ED05A0  SSH1  80C8C3C9  2F41D000  ACTIVE
_   00B79070  PPLX  80E7C3C9  2F41F000  NOT LOCAL HCI
_   00B752A0  E510  80E7C3C9  35404000  NOT LOCAL HCI
_   00B71540  QS50  80E7C3C9  2F439000  NOT LOCAL HCI
_   00F89AD0  X520  80C8C3C9  2ED1F000  ACTIVE
_   00ED0070  X510  80C8C3C9  2ED21000  ACTIVE
_   00F895A0  X530  80C8C3C9  2F6BB000  ACTIVE
_   00B8EAD0  S520  80E7C3C9  2F6AD000  NOT LOCAL HCI
_   00B71010  S510  80E7C3C9  2F667000  NOT LOCAL HCI
_   00F89070  CFX2  80C8C3C9  2F639000  ACTIVE
_   00F850C0  CAA2  80C8C3C9  2F60B000  ACTIVE
```

For each HCI subsystem, the following five values are shown and are described below:

ADDRESS Main storage address of the MVS subsystem control table (SSCT), which is represented by this entry.

NAME Subsystem name.

SSCTSUSE SSCTSUSE field in hexadecimal of the SSCT.

SSCTSUS2 SSCTSUS2 field in hexadecimal of the SSCT.

STATUS Current status of the HCI subsystem. Valid values are:

- **NOT (LOCAL) HCI:** This HCI subsystem is a HCI sysplex subsystem.
- **ACTIVE:** The HCI subsystem is active.
- I**NACTIVE:** The HCI subsystem is not active.
- **PENDING-WTOR:** The HCI subsystem has a pending write-to-operator.
- **ABENDED:** The HCI subsystem is abending or has abended.

Screen Functions

You can perform the following functions from the HCI SUBSYSTEM DISPLAY:

- Cause the data on the screen to be refreshed by pressing the *Enter* key.
- Exit from the HCILOOK facility entirely by pressing the END key (*PF3*).
- Invoke an ISPF command by typing it on the command line and pressing the *Enter* key.
- Display the INTERACTIVE HCI DISPLAY panel for a subsystem by typing ***D*** in the first position of the line containing the subsystem you want to display and pressing the *Enter* key.
- Page down to see more subsystems by pressing the DOWN key (*PF8*).
- Page up to see more subsystems by pressing the UP key (*PF7*).

INTERACTIVE HCI DISPLAY

The INTERACTIVE HCI DISPLAY, shown in Figure 10-3, is the main entry screen for HCILOOK. It contains overview information that is most helpful to casual users. Access this screen by entering a subsystem ID when you are prompted, or by selecting a subsystem ID from the HCI SUBSYSTEM DISPLAY (Figure 10-2 on page 10-3).

Note: Updates to the currently running HCI can be made only on this screen. Even though other screens display the same first six lines, this screen is the only one where you can alter this information.

The INTERACTIVE HCI DISPLAY contains four major sections.

- **Header section:** Comprises the first eight lines on the screen.
- **Control block status section:** Comprises the next eight lines on the screen.
- **Storage allocation section:** Comprises the next seven lines on the screen.
- **Current journal name:** Occupies one screen line.

Figure 10-3. INTERACTIVE HCI DISPLAY Screen

```
-------------------------- INTERACTIVE HCI DISPLAY ---------------------------
COMMAND   ===>
 SYSID: X510 SMFID: CW01       JOB: XPEDH51     STEP: XPEDH51   PROC:
STATUS: D118F09000004B00 DUMP UNIT:              VOL:          CLASS: X
  MASK: 0000000000000000  DUMP PFX:              JOB: XP510     USER: HCIUSER
   GCA: 2ED21000       HCI STARTED: 11/04/10  00:16:33     JRNL LOSS: 0
  ASID: 013C VER: 0205   TIMER INT: 1500     ACB INT: 3000      JESC: $

        CURR  HIGH  MAX          CURR  HIGH  MAX          CURR  HIGH  MAX
   AMB: 2     2     2       CCB: 1     2     32      DCB: 1     2     32
   RCB: 1     2     32      TPT: 25    25    25      UIB: 1     2     32
   RPL: 0     0     0       LUB: 0     0     1       WRE: 1     5     256
   JRE: 0     1     2048    ORE: 0     2     2048    SAA: 0     7     7
   PCB: 6     6     6       TEA: 2     4     128     TSB: 2     2     32
   SAB: 0     1     1       SYN: 0     0     0

               CURRENT   HIGHEST                     CURRENT   HIGHEST
PRIVATE BELOW: 7K        13K          PRIVATE ABOVE: 946K      2946K
COMMON  BELOW: 5K        5K           COMMON  ABOVE: 163K      164K
MODULE  BELOW: 40K       40K          MODULE  ABOVE: 622K      633K
BUFFER  BELOW: 0K        0K           BUFFER  ABOVE: 224K      288K

CURRENT JOURNAL: X2.PROD.MLXW510.ALLPTFS.HCIJRNL1             CMPRS: 0.00%
```

Header Section

The header section contains fields that show most of the HCICNGCA configuration options currently in effect. The same screen header is included on the INTERACTIVE HCI DISPLAY, the REGION DISPLAY, the USER DISPLAY, and the SESSION DISPLAY screens providing the same information on each of the screens. However, you can make dynamic updates to fields in this section *only* from the INTERACTIVE HCI DISPLAY. Changes made to these fields on subsequent screens are ignored.

Following is a description of each of the fields in the header section:

COMMAND Input line onto which all of the usual ISPF commands, such as TSO and SDSF, can be entered. No HCILOOK commands are defined for this field on this screen.

SYSID Shows the current HCI subsystem ID. This input field can be changed. Each time you press the *Enter* key, the screen is updated from the HCI whose subsystem ID appears in this field.

SMFID Shows the SMF system identification of the z/OS on which the HCI is running.

JOB Shows the JOBNAME of the executing HCI.

STEP Shows the STEPNAME of the executing HCI.

PROC Shows the PROCNAME of the executing HCI.

STATUS Shows, in hexadecimal representation, the current settings of the GCSATATS field.

DUMP UNIT Shows the value of the DMPUNIT attribute on the HCICNGCA configuration element unless this value has been changed by a prior invocation of the HCILOOK facility, in which case it shows the current value for DUMP UNIT. You can change this input field by typing a new value in this field and pressing the *Enter* key. This new value remains in effect until changed or until the HCI is brought down and back up, at which time the value specified on the HCICNGCA element is once again in effect.

VOL Shows the value of the DMPVOL attribute on the HCICNGCA configuration element, unless this value has been changed by a prior invocation of the HCILOOK facility, in which case it shows the current value for VOL. You can change this input field by typing a new value in this field and pressing the *Enter* key. This new value remains in effect until changed or until the HCI is brought down and back up, at which time the value specified on the HCICNGCA element is once again in effect.

CLASS Shows the value of the DMPCLAS attribute on the HCICNGCA configuration element, unless this value has been changed by a prior invocation of the HCILOOK facility, in which case it shows the current value for CLASS. You can change this input field and by typing a new value in this field and pressing the *Enter* key. Note that the same rules apply for dynamic changes to these fields as applied to the original element specification. If DUMP UNIT and/or VOL are specified, CLASS must be blank. This new value remains in effect until changed or until the HCI is brought down and back up, at which time the value specified on the HCICNGCA element is once again in effect.

MASK Shows, in hexadecimal representation, the current setting of the journal mask bits, where 0000000000000000 indicates that *no* journal records are to be written, and FFFFFFFFFFFFFFFF indicates that *all* journal records are to be written. You can change this input field by typing a new value for these sixteen hex digits and pressing the *Enter* key. This new value remains in effect until changed or until the HCI is brought down and back up, at which time the value specified on the HCICNGCA element is once again in effect.

DUMP PFX Shows the value of the DMPPFX attribute on the HCICNGCA configuration element, unless this value has been changed by a prior invocation of the HCILOOK facility, in which case it shows the current value for DUMP PFX. You can change this input field by typing a new value in this field and pressing the *Enter* key. This new value remains in effect until changed or until the HCI is brought down and back up, at which time the value specified on the HCICNGCA element is once again in effect.

JOB Shows the current JOBNAME prefix that the HCI uses whenever it submits a TP job. This is an input field and can be changed by overtyping a new value in this field and pressing the *Enter* key. This new value remains in effect until changed or until the HCI is brought down and back up, at which time the value specified on the HCICNGCA element is once again in effect.

USER Shows the current USERID that the HCI uses whenever it cannot obtain a USERID from any other source. You can change this input field typing a new value in this field and pressing the *Enter* key. This new value remains in effect until changed or until the HCI is brought down and back up, at which time the value specified on the HCICNGCA element is once again in effect.

GCA Shows the main storage address of the HCI global communications area, the main control block within the HCI. This field has significance in that whenever you place the cursor under the address portion and press the *Enter* key, control is transferred to the HCI REGION DISPLAY screen.

HCI STARTED Shows the date and time that the currently executing HCI was started.

JRNL LOSS Shows the number of journal records that could not be written due to exhaustion of the JRE control block entries. This number should be 0, indicating no lost records. If this number is greater than 0, it indicates that any journals taken during the time the HCI has been executing may be invalid due to lost records.

ASID Shows, in hexadecimal, the ASID number of the currently executing HCI.

VER Shows the version and modification number of the currently executing HCI.

TIMER INT Shows the current value of the main timer interval for the HCI in hundredths of a second. You can change this input field by typing a new value in this field and pressing the *Enter* key. This new value remains in effect until changed or until the HCI is brought down and back up, at which time the value specified on the HCICNGCA element is once again in effect.

ACB INT Shows the current value of the OPEN ACB interval for the HCI in hundredths of a second. You can change this input field by typing a new value in this field and pressing the *Enter* key. This new value remains in effect until changed or until the HCI is brought down and back up, at which time the value specified on the HCICNGCA element is once again in effect.

JESC Shows the current value of the JES2/JES3 control character that is used in CANCEL commands issued internally by the HCI. You can change this input field by typing a new value in this field and pressing the *Enter* key. This new value remains in effect until changed or until the HCI is brought down and back up, at which time the value specified on the HCICNGCA element is once again in effect.

Control Block Status Section

The control block status section lists each of the major control blocks within the HCI and shows the current number allocated, the highest number that has ever been allocated, and the maximum number that can be allocated.

The 17 control blocks displayed in the control block status section are:

AMB Access method block. Static allocation (CURR=HIGH=MAX), one block for each HCICNAMB element defined in the XML configuration document, plus an optional one automatically defined if INTERNAL=YES is specified.

CCB Conversation control block. Dynamic allocation, one block for each conversation currently active.

DCB Destination control block. Dynamic allocation, one block for each destination opened by TPs, or one block for each conversation if the TP does not open a destination (via CWOPEN).

RCB Region control block. Dynamic allocation, one block for each active TP region.

TPT TP profile table. Static allocation, one block for each HCICNTPT element defined in the configuration assembly.

UIB User interface block. Dynamic allocation, one block for each user registered with the HCI.

RPL Request parameter list. Dynamic allocation, one block for each active LU 2 or LU 6.2 VTAM event.

LUB Logical unit block. Dynamic allocation, one block for each active LU 2 session for which the HCI is the primary LU.

WRE Work request element. Dynamic allocation, one block for each active request for a unit of work to be performed by an HCI task.

JRE Journal request element. Dynamic allocation, one block for each active request for a journal record to be written.

ORE Operator request element. Dynamic allocation, one block for each active request for an operator message to be displayed on the console or returned to the initiator of the request.

SAA Save area above (16M line). Dynamic allocation, one block for each active save area allocated above the 16M line.

PCB Port control block. Static allocation, one block for each HCICNPCB element specified in the configuration assembly.

TEA TCP/IP event area. Dynamic allocation, one block for each active TCP/IP event.

TSB TCP/IP subtask block. Dynamic allocation, one block for each active TCP/IP subtask.

SAB Save area below (16M line). Dynamic allocation, one block for each active save area allocated below the 16M line.

SYN Synonym chain status. Counter only, of the current and highest number of hash table synonym chain entries in any of the HCI hash tables.

Storage Allocation Section

This section contains information concerning main storage use within the HCI. Two numbers are given for each of eight storage types:

- Current allocation
- Highest allocation that this storage type has ever occupied.

All values are specified in kilobytes (K), which has a numeric value of 1024 bytes.

PRIVATE BELOW	Storage in the private area below the 16M line
PRIVATE ABOVE	Storage in the private area above the 16M line
COMMON BELOW	Storage in the common area (SP 241 and SP 245) below the 16M line
COMMON ABOVE	Storage in the common area (SP 241 and SP 245) above the 16M line
MODULE BELOW	Storage occupied by HCI modules (SP 252) below the 16M line
MODULE ABOVE	Storage occupied by HCI modules (SP 252) above the 16M line
BUFFER BELOW	Storage used for I/O buffers below the 16M line
BUFFER ABOVE	Storage used for I/O buffers above the 16M line.

Current Journal Name Section

This section contains only one line that shows the name of the HCI journal currently allocated and the percentage of data compression attained.

Screen Functions

You can perform the following functions from the INTERACTIVE HCI DISPLAY:

- Cause the data on the screen to be refreshed by pressing the *Enter* key.
- Invoke an ISPF command by typing it on the command line and pressing the *Enter* key.
- Change the screen to NOUPDATE mode by typing **NOUPDATE** on the command line and pressing the *Enter* key. Compuware recommends running in NOUPDATE mode to prevent to prevent the accidental changing of attribute values.
- Change the screen to UPDATE mode by typing **UPDATE** on the command line and pressing the *Enter* key.
- Type new values for the SYSID, DUMP UNIT, VOL, CLASS, MASK, DUMP PFX, JOB, USER, TIMER INT, ACB INT and/or JESC fields and press the *Enter* key.
- Cause the current journal to be closed and deallocated and the next journal to be allocated and opened by placing the cursor anywhere under the current journal name and pressing the *Enter* key. A delay of up to the value specified in TIMER INT may occur before the journals actually swap, so do not place the cursor under the journal name if a previously requested swap has not yet occurred. You may press the Enter key with the cursor anywhere else on the screen as many times as you want.
- Cause the HCI REGION DISPLAY to be displayed by placing the cursor anywhere under the address value of the GCA field and pressing the *Enter* key.
- Exit from the HCILOOK facility entirely by pressing the END key (*PF3*).

HCI REGION DISPLAY

The HCI REGION DISPLAY, shown in Figure 10-4 on page 10-9, contains two major sections. The first is the *header section*, which is a duplicate of the first section of the INTERACTIVE HCI DISPLAY (Figure 10-3 on page 10-4), except you cannot change the values shown on this screen. The second is the *region display section* that lists all active regions (address spaces) that contain HCI TPs. Each line of the display represents one HCI

region. Access the HCI REGION DISPLAY by positioning the cursor under the address portion of the GCA field on the HCI INTERACTIVE DISPLAY and pressing the *Enter* key.

Figure 10-4. HCI REGION DISPLAY Screen

```
------------------------------ HCI REGION DISPLAY ------------- Row 1 to 1 of 1
COMMAND   ===>                                                 SCROLL ===> PAGE
 SYSID: HBP1 SMFID: CW01       JOB: EFHRIP0I    STEP: HCI       PROC:
STATUS: D118F09080004100 DUMP UNIT: SYSDA        VOL: TMP900   CLASS:
  MASK: 0000000000000002  DUMP PFX: SDUMP        JOB: EFHRIP    USER: EFHRIP0
   GCA: 2F681000       HCI STARTED: 11/04/10  03:00:25     JRNL LOSS: 0
  ASID: 0062 VER: 0205   TIMER INT: 1500     ACB INT: 3000      JESC: $

  ENTER -D- TO DISPLAY ASSOCIATED USER(S), OR
        -C- TO CANCEL THE ENTIRE TP JOB

   ADDRESS   JOBNAME   STEPNAME PROCSTEP JOBNO    STATUS ASID USERS SYSTEM

_  2E64CD28  EFHRIP0I  HCI                        INACT  0062 7     LOCAL
****************************** Bottom of data ********************************
```

Header Section

The fields in this section are the same as those described previously for the INTERACTIVE HCI DISPLAY in “Header Section” on page 10-4, except you cannot change the displayed values.

Region Display Section

For each active region, the following fields and their values are shown:

ADDRESS Main storage address of the region control block (RCB) represented by this entry

JOBNAME JOBNAME of this HCI region

STEPNAME STEPNAME of this HCI region

PROCSTEP PROC step name of this HCI region

JOBNO JES2/JES3 job number of this HCI region

STATUS Current status of this region. Valid values are:

- **ACTIVE:** If the region has an active cross-memory connection with the HCI.
- **INACT:** If the region has no active cross-memory connection with the HCI.

ASID Hexadecimal value of the ASID of this region

USERS Number of active (registered) users within this region

SYSTEM Indicator whether a local or remote z/OS is executing the task within this region. Valid values are:

- **LOCAL:** Executing on the same z/OS as the HCI.
- **REMOTE:** Executing on another sysplex z/OS.

Screen Functions

You can perform the following functions from the HCI REGION DISPLAY:

- Cause the display to be refreshed by typing a single letter on the command line and pressing the *Enter* key.
- Display the HCI USER DISPLAY screen for a specified region by typing **D** in the first position of the line containing the region you want to display and pressing the *Enter* key.
- Cause a specified region to be canceled by typing ***C*** in the first position of the line containing the region you want to cancel and pressing the *Enter* key. You are prompted to verify that you want to cancel the region. If you want to cancel the region, reply **YES** to the prompt. If you do not want to cancel the region, reply **NO** to the prompt. In either case, you are returned to this screen.
- Display the HCI SYSPLEX DISPLAY screen by typing **SYSPLEX** on the command line and pressing the *Enter* key.
- Display the HCI MODULE DISPLAY screen by typing **MODULES** on the command line and pressing the *Enter* key. A delay of up to one minute may occur before the module display appears.
- Invoke the HCI MEMORY DISPLAY screen by placing the cursor under the address of the GCA field and pressing the *Enter* key.
- Page down to see more subsystems by pressing the DOWN key (*PF8*).
- Page up to see more subsystems by pressing the UP key (*PF7*).
- Return to the INTERACTIVE HCI DISPLAY panel by pressing the END key (*PF3*).

HCI SYSPLEX DISPLAY

The HCI SYSPLEX DISPLAY, shown in Figure 10-5 on page 10-11, contains two major sections. The first is the *header section*, which is a duplicate of the first section of the INTERACTIVE HCI DISPLAY (Figure 10-3 on page 10-4), except that you cannot change the values shown on this screen. If the HCI has been configured for sysplex support, the second section is the *sysplex display section*, which lists all of the z/OS systems in the sysplex. Access this screen by entering the **SYSPLEX** command on the command line of the HCI REGION DISPLAY (Figure 10-4 on page 10-9).

Figure 10-5. HCI SYSPLEX DISPLAY Screen

```
--------------------------- HCI SYSPLEX DISPLAY -------------- Row 1 to 2 of 2
COMMAND   ===>                                              SCROLL ===> PAGE
 SYSID: X510 SMFID: CW01       JOB: XPEDH51     STEP: XPEDH51    PROC:
STATUS: D118F09000004B00 DUMP UNIT:              VOL:          CLASS: X
  MASK: 0000000000000000  DUMP PFX:              JOB: XP510     USER: HCIUSER
   GCA: 2ED21000       HCI STARTED: 11/04/10  00:16:33     JRNL LOSS: 0
  ASID: 013C VER: 0205   TIMER INT: 1500     ACB INT: 3000      JESC: $

HCI CONFIGURED FOR SYSPLEX

   ADDRESS   NAME       TYPE      STATUS    HCIYPREP  COUNT

   3797F828  CW01       LOCAL     ACTIVE    COMPLETE  0
   37975D88  CW09       REMOTE    ACTIVE    COMPLETE  0
****************************** Bottom of data ********************************
```

Header Section

The fields in this section are the same as those described previously for the INTERACTIVE HCI DISPLAY in “Header Section” on page 10-4, except you cannot change the displayed values.

Sysplex Display Section

Each row is a z/OS member of the sysplex and shows the following six fields and their values:

ADDRESS Main storage address of the HCI subsystem block (SSB) represented by this entry

NAME Name of the sysplex member

TYPE Indicator whether the member is local or remote as known to the HCI:

- **LOCAL:** Executing on the same z/OS as the HCI.
- **REMOTE:** Executing on another sysplex z/OS.

STATUS Indicator whether the member is active or inactive with the HCI:

- **ACTIVE:** The HCI will submit jobs to the z/OS.
- **INACT:** The HCI will not submit jobs to the z/OS.

HCIYPREP Indicator whether the HCIYPREP procedure has completed or is pending execution:

- **COMPLETED:** HCIYPREP has completed.
- **PENDING:** HCIYPREP is pending execution.

COUNT Number of of jobs submitted by the HCI via the ROUTE statement in the HCI PARMLIB

Screen Functions

You can perform the following functions from the HCI SYSPLEX DISPLAY

- Cause the display to be refreshed by typing a single letter on the command line and pressing the *Enter* key.

- Page down to see more members by pressing the DOWN key (*PF8*).
- Page up to see more members by pressing the UP key (*PF7*).
- Return to the HCI REGION DISPLAY screen by pressing the END key (*PF3*).

HCI USER DISPLAY

The HCI USER DISPLAY screen, shown in Figure 10-6, contains two major sections. The first is the *header section*, which is a duplicate of the first section of the INTERACTIVE HCI DISPLAY screen, except that you cannot change the values shown on this screen. The second is the *user display section* that lists all active users (registered TPs) for the TP job selected on the HCI REGION DISPLAY screen. Access the HCI USER DISPLAY by entering the D line command next to the desired TP job on the HCI REGION DISPLAY (Figure 10-4 on page 10-9).

Figure 10-6. HCI USER DISPLAY Screen

```
------------------------------ HCI USER DISPLAY -------------- Row 1 to 2 of 7
COMMAND   ===>                                                SCROLL ===> PAGE
 SYSID: HBP1 SMFID: CW01       JOB: EFHRIPOI    STEP: HCI       PROC:
STATUS: D118F09080004100 DUMP UNIT: SYSDA        VOL: TMP900   CLASS:
  MASK: 0000000000000002  DUMP PFX: SDUMP        JOB: EFHRIP    USER: EFHRIPO
   GCA: 2F681000       HCI STARTED: 11/04/10  03:00:25     JRNL LOSS: 0
  ASID: 0062 VER: 0205   TIMER INT: 1500     ACB INT: 3000      JESC: $

  ENTER -D- TO DISPLAY ASSOCIATED CONVERSATION(S), OR
        -U- TO UNREGISTER THE USER

_ LCLTP NAME: P10947
      SYSTEM: *LOCAL*   MEMBER: P10947
         UIB: 2E1A5138  USERID: PFHKSF0   STATUS: WAITING
     TP NAME: P10947
    CPI-C RC: 0         HCI RC: 0         MODULE:

_ LCLTP NAME: P16806
      SYSTEM: *LOCAL*   MEMBER: P16806
         UIB: 2E1A6DA8  USERID: BFHCDM0   STATUS: WAITING
     TP NAME: P16806
    CPI-C RC: 0         HCI RC: 0         MODULE:
```

Header Section

The fields in this section are the same as those described previously for the INTERACTIVE HCI DISPLAY in “Header Section” on page 10-4, except you cannot change the displayed values.

User Display Section

Each user display entry is five display lines in length and shows the following fields and their values::

LCLTP NAME 1- to 64-byte TP name that this user has registered with the HCI.

SYSTEM z/OS system name

MEMBER Name of the HCI PARMLIB member read when the HCI submitted the TP job.

UIB Hexadecimal representation of the main storage address of the user interface block (UIB) from which the information for this entry came.

USERID User ID currently in effect for this TP.

STATUS The current status of this TP:

- **WAITING** - Indicates that the TP has issued a CWWAIT or CWWMUE and is currently waiting for an event to occur that will satisfy this wait.
- **EXECUTING IN TP** - Indicates that the TP is not currently executing an HCI call, but is executing within its own code.
- ***XXXXXX* SUSPENDED** - Indicates that the TP has issued the HCI call (represented by *XXXXXX*) and is currently suspended within the PC_ROUTINE waiting for this call to complete.
- ***XXXXXX* IN TP** - Indicates that the TP is executing within its own code, but the last call made to the HCI is shown (represented by *XXXXXX*).

TP NAME 1- to 64-byte TP_NAME from the HCICNTPT entry

CPI-C RC Numeric value of the last CPI-C return code passed back to the TP from an HCI call.

HCI RC Hexadecimal representation of the last HCI return code passed back to the TP from an HCI call.

MODULE Name of the HCI module that last set a non-zero CPI-C or HCI return code value.

Screen Functions

You can perform the following functions from the HCI USER DISPLAY screen:

- Cause the display to be refreshed by typing a single letter on the command line and pressing the *Enter* key.
- Display the HCI SESSION DISPLAY screen by typing **D** in the first position of the line containing the TP you want to display and pressing the *Enter* key.
- Cause a specified TP to be unregistered by typing **U** in the first position of the line containing the TP you want to unregister and pressing the *Enter* key. You are prompted to verify that you want to unregister this TP. If you want to unregister the TP, reply **YES** to the prompt. If you do not want to unregister the TP, reply **NO** to the prompt. In either case, you return to this screen.
- Page down to see more members by pressing the DOWN key (*PF8*).
- Page up to see more members by pressing the UP key (*PF7*).
- Return to the HCI REGION DISPLAY screen by pressing the END key (*PF3*).

HCI SESSION DISPLAY

The HCI SESSION DISPLAY screen, shown in Figure 10-7 on page 10-14, contains two major sections. The first is the *header section*, which is a duplicate of the first section of the INTERACTIVE HCI DISPLAY (Figure 10-3 on page 10-4), except that you cannot change the values shown on this screen. The second section is the session display section, which is a list of all active sessions for the TP selected from the HCI USER DISPLAY. Access the HCI SESSION DISPLAY by typing **D** in the first position of the line containing the TP you want to display and pressing the *Enter* key on the HCI USER DISPLAY (Figure 10-6 on page 10-12).

Figure 10-7. HCI SESSION DISPLAY Screen

```
---------------------------- HCI SESSION DISPLAY ------------- Row 1 to 1 of 1
COMMAND   ===>                                                SCROLL ===> PAGE
 SYSID: HBP1 SMFID: CW01       JOB: EFHRIPOI    STEP: HCI       PROC:
STATUS: D118F09080004100 DUMP UNIT: SYSDA        VOL: TMP900   CLASS:
  MASK: 0000000000000002  DUMP PFX: SDUMP        JOB: EFHRIP    USER: EFHRIPO
   GCA: 2F681000       HCI STARTED: 11/04/10  03:00:25     JRNL LOSS: 0
  ASID: 0062 VER: 0205   TIMER INT: 1500     ACB INT: 3000      JESC: $


LCLTP NAME: P10947
  ACB NAME: H08B75C0 TYPE: TCP/IP GEN NAME:          APPLID: 10.10.0.200
 PARTNER LU NAME: 10.30.0.94          MODE: :10947/  STATUS: 00000000
        MAX SESS:           CONTENTION WIN:            LOSE:
   RECEIVE COUNT: 19           SEND COUNT: 22        STATE: SEND RECEIVE

****************************** Bottom of data ******************************
```

Header Section

The fields in this section are the same as those described previously for the INTERACTIVE HCI DISPLAY in “Header Section” on page 10-4, except you cannot change the displayed values.

Session Display Section

The session display section displays the following fields once for each currently active session for the selected TP within the selected JOB.

LCLTP NAME 1- to 64-byte TP name that this user has registered with the HCI.

ACB NAME VTAM ACB name that is used for this session for LU 6.2 and LU 2 conversations; or contains the subtask identification used between the HCI and TCP/IP for this connection.

TYPE Type of conversation represented by this screen entry:

- LU 6.2 - For LU 6.2 conversations
- LU 2 - For LU 2 conversations
- TCP/IP - For TCP/IP connections.

GEN NAME VTAM generic name used by the partner LU to establish this session with the HCI for LU 6.2 and LU 2 conversations. This field is blank for TCP/IP connections.

APPLID Application associated with the ACB NAME used by the HCI for this LU 6.2 or LU 2 conversation; or contains the local IP address (not the partner IP address) of this host TCP/IP.

PARTNER LU NAME Name of the partner LU as it is known to the local VTAM for LU 6.2 or LU 2 conversations; or contains the IP address of the remote partner for TCP/IP connections.

MODE Logon mode name used to establish the session over which this conversation is flowing for LU 6.2 conversations. This field is blank for LU 2 conversations. For TCP/IP connections, this field contains the local port number over which the HCI is communicating with TCP/IP. This port number of prefixed with a colon (:) and suffixed with a forward slash (/) to differentiate it from other field values.

STATUS Hexadecimal representation of the status bytes currently set in the conversation control bLock (CCB) within the HCI for this conversation and all conversation types.

MAX SESS Value specified in the HCI configuration or in the VTAM application definition statement for maximum sessions. This field is blank for LU 2 and TCP/IP.

CONTENTION WIN Current number of contention winner sessions allocated to this mode name group for LU 6.2 conversations. This field is blank for LU 2 and TCP/IP.

LOSE Current number of contention loser sessions allocated to this mode name group for LU 6.2 conversations. This field is blank for LU 2 and TCP/IP.

RECEIVE COUNT Number of messages received by the HCI for this conversation for all conversation types.

SEND COUNT Number of messages sent by the HCI for this conversation for all conversation types.

STATE Current state of the conversation:

- NULL
- RESET
- INITIALIZED
- SEND
- RECEIVE
- SEND PENDING
- CONFIRM
- CONFIRM SEND
- CONFIRM DEALLOC
- DEFER RECEIVE
- DEFER DEALLOC
- SYNC POINT
- SYNC POINT SEND
- DEAL PEND CONF
- RECV PEND CONF
- SEND PEND CONF
- SEND RECEIVE (TCP/IP only)

Refer to IBM's CPI-C programming manual for a definition of the state values.

Screen Functions

You can perform the following functions from the HCI SESSION DISPLAY screen:

- Cause the display to be refreshed by typing a single letter on the command line and pressing the *Enter* key.
- Page down to see more members by pressing the DOWN key (*PF8*).
- Page up to see more members by pressing the UP key (*PF7*).
- Return to the HCI USER DISPLAY screen by pressing the END key (*PF3*).

HCI MEMORY DISPLAY

The HCI MEMORY DISPLAY, shown in Figure 10-8, displays storage from z/OS common areas and from the HCI private region. Storage from private regions other than the HCI cannot be displayed. Invoke this screen from the HCI REGION DISPLAY (Figure 10-4 on page 10-9) by placing the cursor under the GCA address field and pressing the *Enter* key.

Figure 10-8. HCI MEMORY DISPLAY Screen

```
------------------------------ HCI MEMORY DISPLAY ---------- Row 1 to 18 of 256
COMMAND   ===>                                                SCROLL ===> PAGE
 SYSID: HBP1    STORAGE LENGTH: 001000    NEXT ADDR: 2F682000
                                                GCA: 2F681000

 OFFSET ------------------------------------------------------------   ADDRESS
 000000  C7C3C140  00000000  C6D41A4C  3EB46D01 * GCA ....FM...... *  2F681000
 000010  00000000  00000000  C8C2D7F1  C3E6F0F1 * ........HBP1CW01 *  2F681010
 000020  C5C6C8D9  C9D7F0C9  C8C3C940  40404040 * EFHRIP0IHCI      *  2F681020
 000030  40404040  40404040  D118F090  80004100 *          J.0..... *  2F681030
 000040  E2E8E2C4  C1404040  E3D4D7F9  F0F0405B * SYSDA   TMP900 . *  2F681040
 000050  40404040  000005DC  C8C3C9E8  D7D9C5D7 *      ....HCIYPREP *  2F681050
 000060  E2C4E4D4  D7404040  40404040  00000000 * SDUMP       .... *  2F681060
 000070  0000F200  00000000  00000000  00000002 * ..2............. *  2F681070
 000080  00000000  00000000  00000000  00000000 * ................ *  2F681080
 000090  00000000  000002D3  7F006788  C5C6C8D9 * .......L....EFHR *  2F681090
 0000A0  C9D74040  00000208  00000000  00010000 * IP  ............. *  2F6810A0
 0000B0  00035D0A  20988000  00000000  00000000 * ................ *  2F6810B0
 0000C0  00000000  00000000  00000000  F0F2F0F5 * ............0205 *  2F6810C0
 0000D0  40404040  40404040  00000000  00000001 *         ........ *  2F6810D0
 0000E0  00000000  00000001  00FB1000  008E59C0 * ................ *  2F6810E0
 0000F0  008E5938  00000000  00000062  37974000 * ............... . *  2F6810F0
 000100  00000000  00000000  008B87A0  00007078 * ................ *  2F681100
```

Header Section

The header section of the HCI MEMORY DISPLAY differs from the previous screens. The following four fields and their values are displayed:.

SYSID — Subsystem ID of the currently displayed HCI.

STORAGE LENGTH — Hexadecimal representation of the number of bytes that are displayed and that can be browsed.

NEXT ADDR — Hexadecimal representation of the main storage address computed by adding the current main storage address with the storage length.

GCA — Main storage address of the GCA.

Memory Display Section

The Memory Display Section displays in hexadecimal and character representation the contents of main storage starting at the address initially selected. Sixteen bytes of memory are displayed on each line as follows:

OFFSET — Hexadecimal representation of the offset from the initial starting address that this line of the display represents.

Hex/Char Display — Hexadecimal and character representation of the data at the current offset location.

ADDRESS — Hexadecimal representation of the main storage address of the first byte of data on the current line of the display.

Screen Functions

You can perform the following functions from the HCI MEMORY DISPLAY screen:

- Page down to see more members by pressing the DOWN key (*PF8*).
- Page up to see more members by pressing the UP key (*PF7*).
- Display different main storage by placing the cursor under the field that contains the address to be displayed and pressing the *Enter* key.
- Return to the HCI REGION DISPLAY screen by pressing the END key (*PF3*).

HCI MODULE DISPLAY

The HCI MODULE DISPLAY, shown in Figure 10-9, lists all modules within the HCI region. To invoke this screen, enter **MODULES** on the command line of the HCI REGION DISPLAY (Figure 10-4 on page 10-9). A delay of up to a minute can occur before the display appears so that the execution of the HCILOOK facility does not impact the HCI execution.

Figure 10-9. HCI MODULE DISPLAY Screen

```
------------------------------ HCI MODULE DISPLAY ---------- Row 1 to 11 of 248
COMMAND   ===>                                                SCROLL ===> PAGE
 SYSID: HBP1 SMFID: CW01       JOB: EFHRIP0I    STEP: HCI       PROC:
STATUS: D118F09080004100 DUMP UNIT: SYSDA        VOL: TMP900   CLASS:
  MASK: 0000000000000002  DUMP PFX: SDUMP        JOB: EFHRIP    USER: EFHRIP0
   GCA: 2F681000       HCI STARTED: 11/04/10  03:00:25     JRNL LOSS: 0
  ASID: 0062 VER: 0205   TIMER INT: 1500     ACB INT: 3000      JESC: $

  ENTER -D- ON THE MODULE LINE TO DISPLAY THE MODULE


       NAME      ADDRESS     LENGTH      DATE      TIME     VERSION
-------------------------------------------------------------------------
_    DYNAMAIN    02DB6950    002ED8    09/07/05    09.18    01 00 00
_    HCIAACSC    3799DC58              20030804    08.57    02 05 00
_    HCIAACSI    3799A058              20030804    08.57    02 05 00
_    HCIAALCC    3799D5E8              20030804    08.57    02 05 00
_    HCIAALCI    379A1DB8              20030804    08.57    02 05 00
_    HCIAATNM    3799C038              20030804    08.57    02 05 00
_    HCIABEND    3799B000              20030804    08.57    02 05 00
_    HCIACFMD    379A9018              20030804    08.59    02 05 00
_    HCIACNFC    3799E408              20030804    08.59    02 05 00
_    HCIACNFI    3799D118              20030804    08.59    02 05 00
```

Header Section

The fields in this section are the same as those described previously for the INTERACTIVE HCI DISPLAY in “Header Section” on page 10-4, except you cannot change the displayed values.

Module Display Section

The modules display section displays the following fields:

NAME HCI module name

ADDRESS Hexadecimal representation of the main storage address of the HCI module.

LENGTH Hexadecimal representation of the length of the module (CSA modules only).

DATE	Date that the module was last assembled.
TIME	Time that the module was last assembled.
VERSION	Version, release, and modification level of this module.

Screen Functions

You can perform the following functions from the HCI MODULE DISPLAY screen:

- Display the contents of a specific module by typing **D** to the left of the module name on the line containing the desired module and pressing the *Enter* key.
- Page down to see more members by pressing the DOWN key (*PF8*).
- Page up to see more members by pressing the UP key (*PF7*).
- Return to the HCI USER DISPLAY screen by pressing the END key (*PF3*).

Export Facility

The export process reads an HCI journal ESDS and writes a fixed length, compressed, sequential dataset that can easily be transmitted using FTP. Ensure that you specify **BIN** in the FTP command stream.

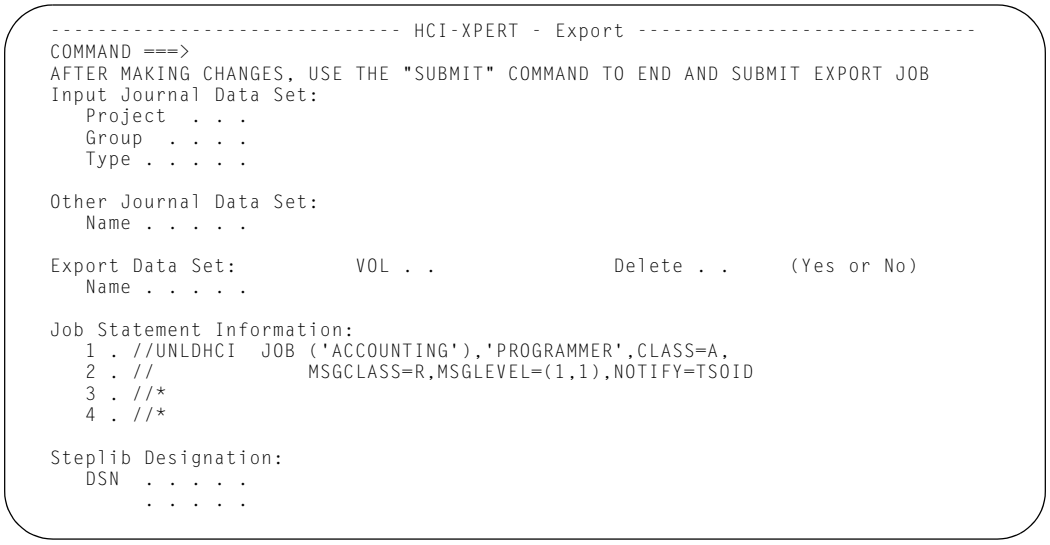
HCI-XPERT - Export Screen

The HCI-XPERT - Export screen, shown in Figure 10-10 on page 10-19, lets you specify the HCI journal to be exported and the exported dataset name. In addition, the following can also be specified:

- Job statement
- Optional STEPLIB libraries
- Deletion of the exported dataset, if already existing
- Volume serial of the exported dataset, if required
- Miscellaneous job statements, such as the OUTPUT statement.

Access this screen by entering **3** (Export) on the command line of the HCI-XPERT main menu (Figure 10-1 on page 10-2).

Figure 10-10. HCI-XPERT - Export Screen



Field Descriptions

The HCI-XPERT - Export screen contains the following fields:

Input Journal Data Set — Standard ISPF method of VSAM ESD dataset specification.

Other Journal Data Set — Alternative input field for instead of the standard ISPF method above, a fully qualified dataset enclosed in single quotation marks or a dataset name without quotation marks to be prefixed with the user's TSO ID.

Export Data Set — Name of the export dataset to be created as a fully qualified dataset enclosed in single quotations marks, or an unqualified to be prefixed with the user's TSO ID.

VOL — VOLSER of the export dataset if required.

Delete — If an export dataset already exists under the same name, specify Y to delete it and create a new one.

Job Statement Information — Your TSO job card and any other required job statements.

Steplib Designation — STEPLIB datasets if required.

Screen Functions

You can perform the following functions from the HCI-XPERT - Export screen:

- Refresh the data on the screen by pressing the *Enter* key.
- View the JCL to be submitted by entering **SHOW** on the command line. This command does not submit the export job or exit the screen.
- Submit the export job and exit the screen by entering **SUBMIT** on the command line
- Exit from this screen without submitting the export job by pressing the END key (*PF3*).

Batch Job Submission

As an alternative to the online Export facility, you can use the JCL example shown in Figure 10-11 to export the journal.

Figure 10-11. JCL to Export an HCI Journal

```
//HCIXPRT JOB ('accounting'),'programmer',CLASS=a,
//         MSGLEVEL=(1,1),NOTIFY=&SYSUID
//EXPORT EXEC PGM=HCIJUNLD
//STEPLIB  DD DSN=HC.MLHCnnn.SLHCLOAD,DISP=SHR
//HCIINDD  DD DSN=*** hci journal vsam ESDS ***
//HCIOUTDD DD DSN=EXPORTED.HCI.JOURNAL,DISP=(,CATLG),UNIT=SYSDA,
//            SPACE=(TRK,(270,150),RLSE),VOL=SER=volser,
//            DCB=(RECFM=F,LRECL=4160,BLKSIZE=4160)
```

You can transmit the dataset named EXPORTED.HCI.JOURNAL in this job to Compuware by FTP. Ensure that you specify **BIN** in the FTP command stream.

Messages and Codes Facility

The Messages and Codes Facility lets you view a designated message and related information. Information displayed for the message includes:

- Message text (@@ indicates data substitution)
- Description of message
- System action
- User action
- Miscellaneous information.

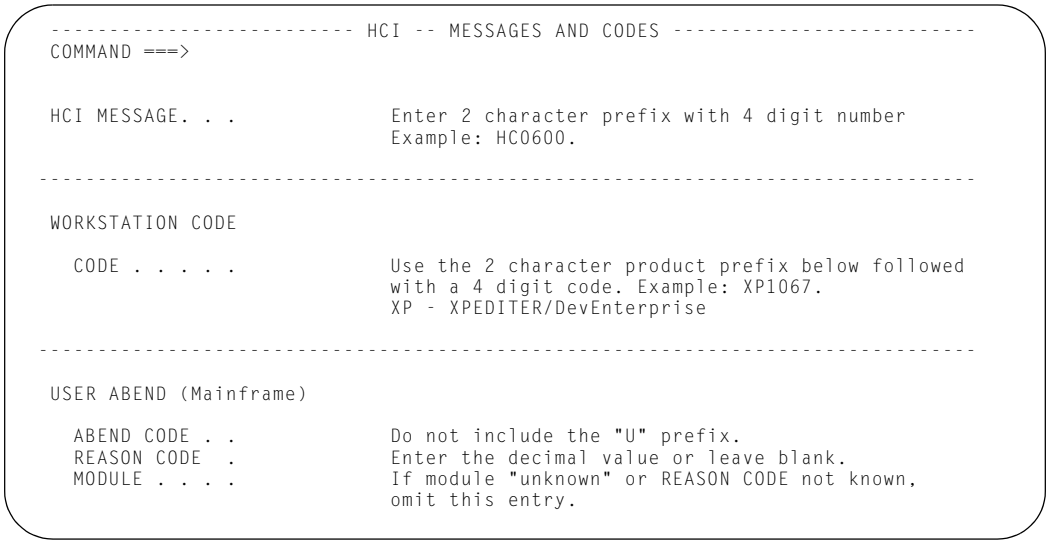
HCI -- MESSAGES AND CODES Screen

The HCI -- MESSAGES AND CODES screen, shown in Figure 10-12 on page 10-21, consists of the following three sections:

- HCI messages
- Workstation codes
- User abends from the HCI.

Each section is mutually exclusive of the other. You can view messages of only one category at a time. Access this screen by entering **5** (Messages) on the command line of the HCI-XPERT main menu (Figure 10-1 on page 10-2).

Figure 10-12. HCI -- MESSAGES AND CODES Screen



HCI MESSAGE Section

The HCI MESSAGE section contains one field:

HCI MESSAGE Enter the message number. The message is usually found on the system console as HCIHC*xxxxs*. Omit the prefix HCI and the suffixed one-character severity code *s* and enter **HCxxxx**.

WORKSTATION CODE Section

The WORKSTATION CODE section contains one field:

CODE Enter the workstation code as a four-digit number. Prefix this with the two-character product code as designated on the screen.

USER ABEND (Mainframe) Section

The USER ABEND (Mainframe) section contains the following three fields:

ABEND CODE Enter the user abend code, omitting the U prefix and any leading zeroes.

REASON CODE If there is a reason code, enter it here in its decimal value.

MODULE If there is a module name, enter it here. If the reason code has been left blank, omit this entry.

Screen Functions

You can perform the following functions from the HCI MESSAGES AND CODES screen:

Once you have displayed the desired message, you can enter the following primary commands on the command line:

- **PREVIOUS:** Positions the screen backward to the previous message.

- **NEXT:** Positions the display forward to the next message.

Once you have displayed the desired next abend code, you can enter the following additional primary commands on the command line:

- **UP:** Moves the screen backward within abend code information.
- **DOWN:** Moves the screen forward within the abend code information.

At any time, you can exit the screen by pressing the END key (*PF3*) or by entering the END command on the command line.

Message Examples

Figure 10-13 shows HCI message HC0600 with installation notes.

Figure 10-13. HCI Message Example

```
-------------------------- HCI -- MESSAGES AND CODES --------------------------
COMMAND ===>                                                  SCROLL ===>

                                                              MORE: +
MSG ID: HCIHC0600I               SEVERITY: INFORMATION

TEXT: @@ TCP/IP @@ ERROR. RETCODE=@@ ERRNO=@@

Explanation: For the indicated Subsystem ID, TCP/IP has returned a negative
value, indicating an event failure. The name of the event, the return code and
the error number are given in this message.

System Action: Processing continues.

User Action: Refer to the TCP/IP Messages and Codes manual for an explanation
of the error message number.

Install Note for TCP/IP release 3.1.0: Insure PTF UN97857 is on.

Install Note for TCP/IP release 3.1.1: Insure PTF UN97858 or higher is on.

Install Note for TCP/IP release 3.2.0: Insure PTF UQ19522 is on.
```

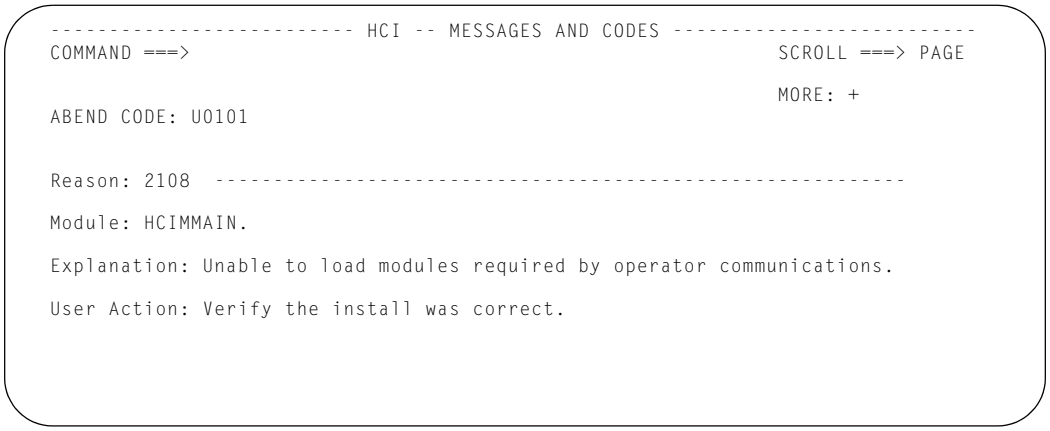
Figure 10-14 shows HCI workstation message XP1084.

Figure 10-14. HCI Workstation Message Example

```
-------------------------- HCI -- MESSAGES AND CODES --------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

                                                              MORE: +
CODE ID: XP1084                  SEVERITY: ERROR

ABSTRACT: THE PROCESS OF STARTING TP ON A MAINFRAME HAS TIMED OUT

System Action: TP fails to connect.

User Action: Perform the following checks:
1. Make sure PARMLIB is not archived and is accessable.
2. Check member name /TP Name association in HCI Configuration (HCICNFIG)
file.
3. Make sure member name association with this TP exists in PARMLIB.
4. If using SYSPLEX, check for HC1306W message in HCI job, which usually
indicates that one of the MVS systems in a SYSPLEX rejects "ROUTE
xxxx,START..." command if it was not previously authorized for HCI through
security.
```

Figure 10-15 shows HCI user abend U0101, reason code 2108, and module HCIMAIN.

Figure 10-15. HCI User Abend Example



Appendix A.
Sample XML Parameters

Figure A-1 and Figure A-2 show a sample XML document that exists within the //HCIXML DD dataset.

Figure A-1. Sample XML Parameters Document (Part 1 of 2)

```
<HCICNFIG>
    <HCICNGCA
        SYSID="AWC6"
        DAE="NO"
        DFLTUSR="DFLTUSR"
        DMPCLAS=""
        DMPPFX="SDUMP"
        DMPUNIT="SYSDA"
        DMPVOL="TMP900"
        EOM="NO"
        INTERNAL="YES"
        JESCHAR=”$”
        JESID="2"
        JESJPRM="NO"
        JRNMASK="FFFFFFFFFFFFFFFF"
        MAININT="1500"
        MAXCCBS="40"
        MAXDCBS="32"
        MAXJRES="4095"
        MAXLUBS="1"
        MAXRCBS="16"
        MAXUIBS="32"
        MAXWRES="2048"
        OACBINT="3000"
        OPCMD="MODIFY"
        PACING="0"
        PREPROC="HCIYPREP"
        ROUTCMD="YES"
        SECFAC=""
        SRVRJOB="HCIJOB"
        SYSPLEX=""
        TCPAMSG="NO"
        TPSEC="NO"
    />
    <HCICNAMB
        LUTYPE="TCPIP"
        MODEL="0"
        PGMNAME="HCITMAIN"
        RCVLN="32760"
        ACBNAME="H08B35D0"
        GENNAME=""
        IPV6="NO"
        HPNS="CURR"
        TCPNAME="TCPIP01"
        TCPPSWD=""
    />
```

Figure A-2. Sample XML Parameters Document, Continued (Part 2 of 2)

```
    <HCICNJCB
            DDNAME="JOURNAL1"
            DSNAME="HCIJOB.HCI.CLIENT.CW01.JOURNAL1"
        />
        <HCICNJCB
            DDNAME="JOURNAL2"
            DSNAME="HCIJOB.HCI.CLIENT.CW01.JOURNAL2"
        />
        <HCICNJCB
            DDNAME="JOURNAL3"
            DSNAME="HCIJOB.HCI.CLIENT.CW01.JOURNAL3"
        />
        <HCICNSIT
            SYMNAME="IPSERVER"
            LUTYPE="TCPIP"
            PARSESS="YES"
            BYPDLAY="NO"
            PLUNAME=""
            MODNAME=""
            LLUNAME=""
            TCPNAME="TCPIP01"
            PORT="16458"
            USERID=""
            PASSWD="********"
            MINCW="0"
            MINCL="0"
            MAXS="0"
            IPNAME="PARTNER.IP.NAME"
            TPNAME=""
        />
        <HCICNTPT
            MEMBER="HCIREQ01"
            CATEGORY=""
            SRVRJOB=""
            MAXCNV="1"
            MAXTPS="999"
            MAXPTIM="1440"
            MAXSTIM="1440"
            USERSEC="YES"
            MULTSEC="NO"
            WILDCRD="NO"
            RECVLEN="32760"
            DSNAME=""
            ALIASOF=""
            TPNAME="HCIREQ01"
        />
    </HCICNFIG>
```

Index

A

B

C

D

E

F

H

I

J

M

O

P

R

S

T

U

V

X