Honeywell

Computer Gateway User Manual

CG11-410

Total Plant

Honeywell

Implementation
Computer Gateway

Computer Gateway User Manual

CG11-410
4/96

Copyright, Trademarks, and Notices

Revision 03 - April 30, 1996

About This Publication

NOTE

The contents of this manual are needed by only those who intend to develop their own interface between a computer and TDC 3000X. If your TDC 3000X System includes the CM50S or CM50N, that interface is already provided and you should not need this manual.

This publication provides information that will aid you in the development of the software necessary to provide the bridge between your host processor and the TDC 3000X process control system through the Computer Gateway. It does not, by itself, cover everything you need to know to develop your application software. Topics covered elsewhere include:

- TDC 3000X System Concepts
- TDC 3000X System Planning
- TDC 3000X System Configuration

A list of related publications is in paragraph 1.6 of this publication.

This publication supports TDC 3000X software release 400. Please see heading 1.5 for information about important functional changes effective with R400.

Change bars are used to indicate paragraphs, tables, or illustrations containing changes that have been made to this manual effective with the release 400. Pages revised only to correct minor typographical errors contain no change bars.

Table of Contents

Table of Contents

Table of Contents

Table of Contents

INTRODUCTION
Section 1

This section discusses the CG role in a TDC 3000X System; reviews the most significant hardware and software components of the CG; and lists the other publications you need to consult during implementation and operation of a host computer-to-CG data link.

1.1 COMPUTER GATEWAY ROLE IN TDC 3000X SYSTEMS

Figure 1-1 shows an overview of the TDC 3000X System architecture. The Computer Gateway is a fully integrated node of that system, enabling it to exchange information with all other nodes on the same LCN. Its function is to serve as a communication link between your host computer and the LCN, making data exchanges with the host computer appear identical to those with any other LCN node. These characteristics provide the foundation for a broad range of potential applications for the host computer.

1.2 CG ARCHITECTURE

The Computer Gateway is a standard LCN node. Its hardware components include an LCN interface, MCPU, memory, a power supply, and a Communications Line Interface (CLI) board with an RS-232C compatible or RS-422 compatible data link. An enhanced version of the CG called a Plant Network Module (PLNM) replaces the CLI board with a Computer Network Interface (CNI) board that enables it to communicate with nodes on a DECnet network. The CG memory contains the standard TDC 3000X node environment software along with CG-specific application software and a user-defined database. Its relationships to the Host Processor and to the LCN are shown in Figure 1-2.

1.3 CG PREPARATION

The Computer Gateway is prepared for its role through the system-configuration process, much like any other LCN-resident node. During configuration, the process engineer informs the system that the CG exists, and what options (such as data link type) are selected, then initiates its loading. Then comes point configuration of ACIDP and CRDP data points and their custom data segments.

Final CG preparation, such as the connection of host-processor-resident programs to ACIDPs for scheduling, and the creation of data-point-specific Internal Data Definition Tables are handled through a series of messages initiated by the host processor.

1.4 HOST PREPARATION

The host computer must have the hardware and software to support your choice of either Bisynch or HDLC communications over RS-232C or RS-442 circuits. Additional software is required to support the communications-message structure defined in this document and the specifics of your desired application.

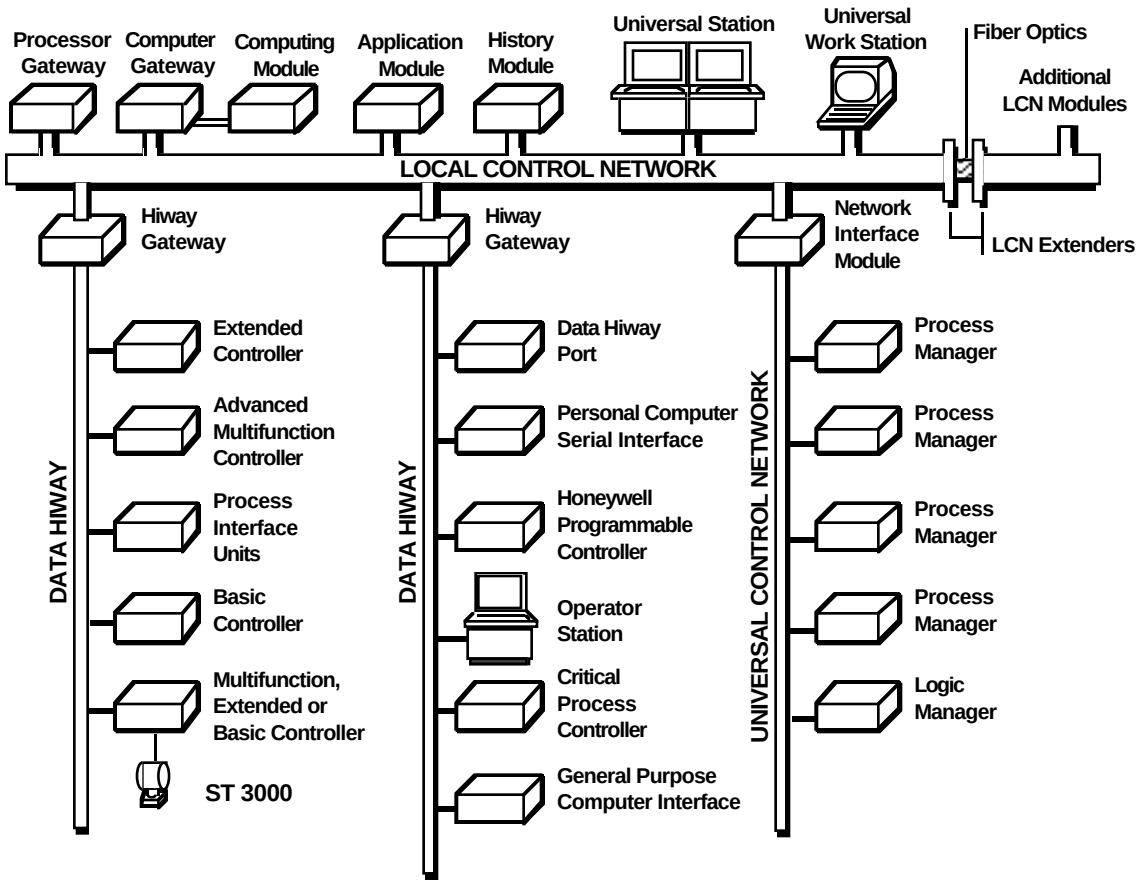
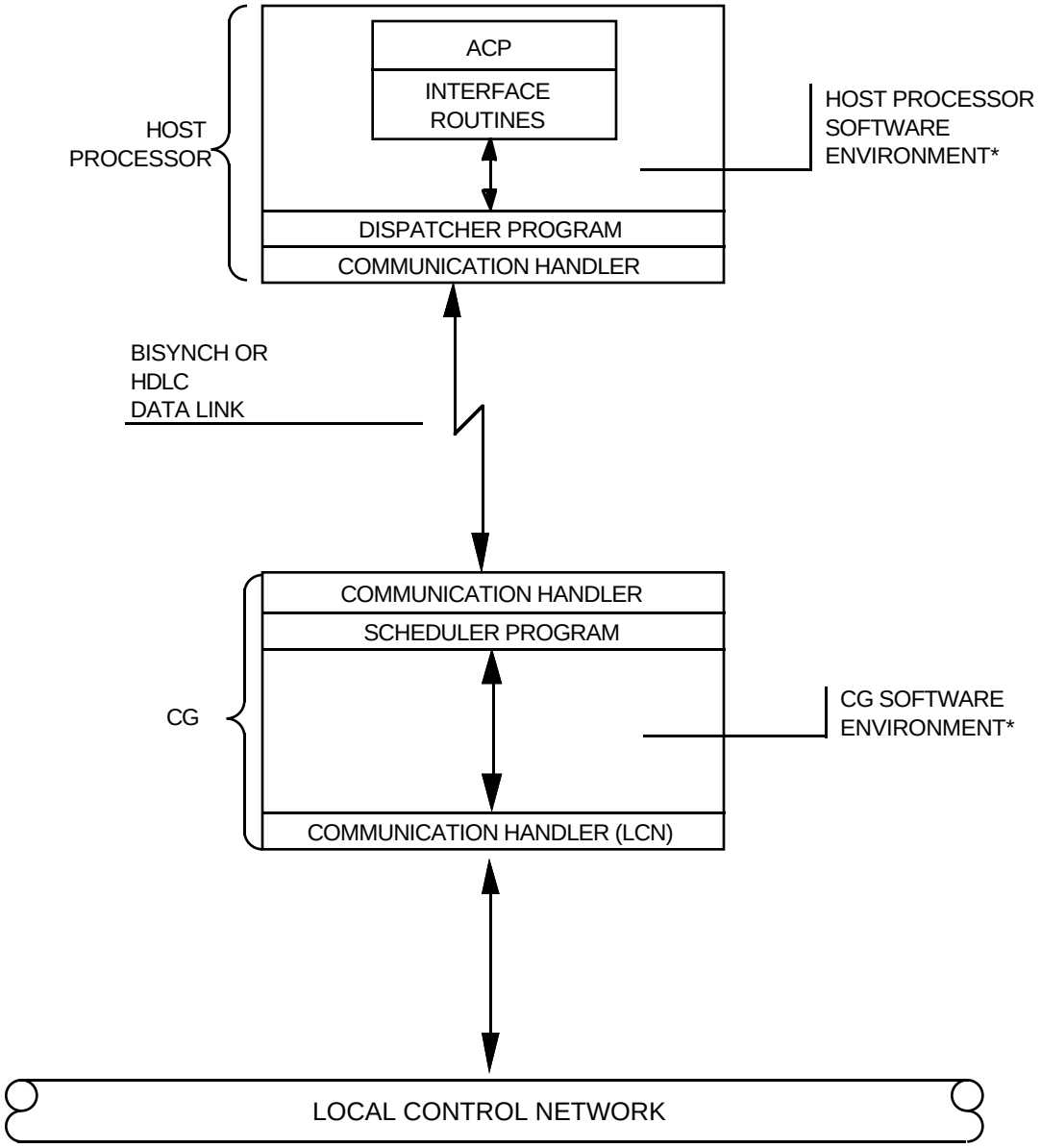


Figure 1-1 — TDC 3000X Architecture

1.5 R400 DIFFERENCES

There are five important changes to the CG at R400.

- Both 8- and 16-character tag names are supported. See subsection 2.3.2 for details.
- Internetwork transfers are supported. See subsection 2.3.3 for details.
- Get History supports 5-, 10- and 20-second history data gathering intervals. The set of returned status codes for Get History has been revised. See subsection 4.8 for details.
- File transfers between the host processor and the LCN History Module (HM) are supported. See subsections 2.1.5 and 4.10 for details.
- Advanced continuous control is supported between the CG and the process connected slots by addition of the RCASENB parameter to ACIDP points. See subsection 6.4 for more information.



* Software environments include all databases, data access mechanisms, and other essential software.

Figure 1-2 — Host Processor/CG/LCN Relationships

1867

1.6 REFERENCES

The following TDC 3000X publications contain additional information related to functions of the CG.

Title	Publication No.	Binder
Computer Gateway Specification and Technical Data	CG03-400	2037
LCN Site Planning Manual	SW02-400	LCN Site Planning
LCN System Installation Manual	SW20-400	LCN Site Installation
LCN System Checkout Manual	SW20-410	LCN Site Installation
Messages Directory	SW09-407	Implementation/Startup & Reconfiguration - 1
Engineer's Reference Manual	SW09-405	Implementation/Startup & Reconfiguration - 2
System Control Functions	SW09-401	Implementation/Startup & Reconfiguration - 2
Computer Gateway Forms	CG88-400	Implementation/ Configuration Forms
Data Entity Builder Manual	SW11-411	Implementation/Engineering Operations - 1
Utilities Operation	SW11-407	Implementation/Engineering Operations - 1
Picture Editor Reference Manual	SW09-450	Implementation/Engineering Operations - 2
Application Module Control Functions	AM09-402	Implementation/AM - 1
Control Language/AM Reference Manual	AM27-410	Implementation/AM - 2
Control Language/AM Data Entry	AM11-485	Implementation/AM - 2
Computer Gateway Form Instructions	CG12-400	Implementation/ Computer Gateway
Computer Gateway Parameter Reference Dictionary	CG09-440	Implementation/ Computer Gateway

FUNCTIONAL OVERVIEW
Section 2

This section shows how the CG works with the host processor to enable your application programs to interact with the various modules on the TDC 3000X Local Control Network (LCN).

2.1 WHAT THE HOST PROCESSOR CAN DO

The following paragraphs introduce you to the types of work the host computer can do by working through the CG to access data from, and send data to, devices on the TDC 3000X Local Control Network. Additional detail on specifics of these facilities is found in later sections.

NOTE

This manual uses the abbreviation ACP to stand for any application program at the host processor that communicates with the CG by means of the various data-link messages.

2.1.1 TDC 3000X Data Access

Before starting an explanation of how the host processor can get and give LCN data, a short explanation of the TDC 3000X database organization and data access is appropriate.

Because the TDC 3000X is a distributed system, its data is distributed among the various nodes. Each piece of data is assigned to a data owner (program) in the node where it resides. When data is addressed by its "external address"—a character-string name in the form `point_name.parameter`—a request is broadcast to all nodes on the LCN. The data owner responds with a numeric "internal address" that can be used for direct access to the data.

Each time an ACP requests a piece of data by its "external address," the "internal address" first must be obtained. Time and LCN loading is saved by having the ACP obtain, and save for reuse, the "internal addresses" for those data points that are referenced regularly. This is automatically done during the building of the Data Definition Tables used in multipoint data accesses.

Current process values are obtained from the CG, the Applications Module, the Hiway Gateway (data from Hiway boxes), or the Network Interface Manager (data from UCN boxes). Historized process values are obtained from the History Module.

2.1.2 Get and Store Current Process Values

Application programs in the host processor can request the input of values from named data point parameters from anywhere on the LCN. These requests can be either for single points or for groups of points specified in prebuilt tables. The host processor can also request the output of values to named data point parameters anywhere on the LCN or on attached Data Hiways. Output requests also can be either for single values or for groups of values and are access controlled to assure system security.

2.1.3 Get History Values

Groups of stored History values can be gathered from any History Module on the LCN. These history values can be either time-based averages or snapshots. Each host request for history values specifies beginning and ending times, and the history type. Prebuilt tables are used to hold the names of the set of point parameters being requested.

2.1.4 Send and Receive Messages

An application program can send character-string messages to all Universal Stations monitoring the operating area to which it is currently assigned through its association with a data point in the CG. At the Universal Stations the message is included in the Message Summary Display and can also be selected for printing and/or journaling. An option to wait for operator confirmation is provided.

Other devices on the LCN can send character-string messages to individual application programs in the host processor. These messages are received by the CG and held, pending a transfer request by the application program. Presence of a pending message at the CG is indicated to the application program in its activation-status information.

2.1.5 File Transfers between Host and HM

Application programs in the host can both read and write files on the LCN History Module. Support functions provide services such as file copy, file renaming, creation, and deletion of directories in user-created volumes, deleting files in user-created volumes, and moving files from one directory to another. Also available are utilities for retrieving volume and file attributes and the listing of volumes and listing of file attributes. See heading 4.10 for details.

File operations are performed at a low priority level. The CG controls the speed of file operations to insure that they do not interfere with higher priority operations.

2.1.6 Role of Data Definition Tables

The most efficient method of data exchange between the host processor and the LCN is by the use of table-driven multipoint value transfers. This is the only method provided for retrieval of history data (current LCN data can be read or written one point at a time as well as by multipoint transfers). As many as four DDTs may get or store data at the same time.

Preparation of the Data Definition Tables (DDTs) is done at the host processor, starting with the entry of lists containing external point-parameter names (organized by data type if for current values). These external names are transferred to the CG for conversion into the internal identifiers required by the LCN and then are stored by the host processor for use in the actual data exchanges. Copies of up to 40 selected input data tables can be held by the CG to save the otherwise repetitive table transfer from host to CG each time the table is used. These CG-resident tables are required for input data precollection.

2.2 HOW THE CG HELPS

The following paragraphs introduce you to the features of the CG that help the host processor accomplish its tasks.

2.2.1 Application Program Activation

The CG activates your application programs (ACPs) in the host processor (by use of a data-link message) in response to the following event types:

- Time of day (Periodic Scheduling)
- Elapsed time since last activation (Cyclic Scheduling)
- Operator demand
- Process special event

The CG activates programs in the host processor on timed and/or demand basis as determined by configuration entries. There are five configuration choices for CG scheduling of host-processor ACPs: cyclic, periodic, demand, cyclic/demand, and periodic/demand. Activation by the CG of host-processor programs can be inhibited by the operator.

All CG scheduling of ACPs requires that the program to be scheduled is "attached" to an ACIDP. See paragraph 2.2.5 for an explanation of the ACIDP and its uses.

ACP activation may be done by the host processor itself, but some restrictions apply. See paragraph 2.3.1.

2.2.1.1 Scheduled Program Activation

Periodic programs first run at a specified daily start time (STIME) and thereafter run at a specified time interval (RTPERIOD).

Example 1: RTPERIOD = 24:00:00 STIME = 17:00:00
This periodic program will run each day at 17:00:00 hours.

Example 2: RTPERIOD = 08:00:00 STIME = 07:00:00
This periodic program will run each day at the following hours: 07:00:00, 15:00:00 and 23:00:00.

Cyclic programs in Normal state first run on host processor start or restart (if their ACIDPs are built with RUN INIT = ON) and thereafter run at a specified time interval (RTPERIOD).

Example 3: RTPERIOD = 00:10:00
Once initiated, this cyclic program will run every 10 minutes.

Scheduled operation of an ACP is not requested by the CG while the associated ACIDP shows it in either Test or Restricted operating state. It must be installed and in Normal state.
The time-interval range for both periodic and cyclic programs is 10 seconds to 24 hours (the CG scheduler function runs at 10-second intervals). The subcategories of periodic/demand and cyclic/demand programs also allow for activation by process-operator demand from the Universal Station.

2.2.1.2 Program Activation by Operator Demand

An ACP can be activated from a Universal Station through the Operator Demand target on its ACIDP parameter display, if its activation type is either demand, cyclic/demand, or periodic/demand. You can also create custom displays that provide for operator-demand activation of ACPs. See the *Picture Editor Reference Manual* for details.

2.2.1.3 Program Activation by Process Special Event

The associated ACP is immediately activated following a store data of ON to an ACIDP's PPS parameter from another ACP, an HG (Event Initiated Processing), or an AM (CL "Set" statement). This activation method is also used by the CG upon receipt of a Message to an ACP. PPS activation is independent of the configured scheduling method.

2.2.1.4 Inhibit of Program Activation

The process operator can prevent/permit an ACP's activation by the CG from the associated ACIDP's Detail Display at the Universal Station. Optionally, you can construct a custom display that performs this function by a store of INHIBIT/PERMIT to the ACDIP INH_STAT parameter.

2.2.2 Process Data Exchanges

The CG receives data-exchange requests (get or store) from the host processor, performs any necessary reformatting and external-to-internal identification translation, transmits the request over the LCN, then returns the data and/or status information to the host processor. Two types of data requests are supported, single-point and table-driven. Typically, single-point requests are used when only a few point values are required; while table-driven requests are used when larger volumes of data are required.

NOTE

Other nodes on the LCN cannot directly access data in the host processor; however, they can read data points stored in the CG database.

2.2.3 History Data Acquisition

The CG receives history-gathering requests from the host processor, performs any necessary reformatting, transmits the request over the LCN, then returns the history data and status information to the host processor. All history-data acquisition calls are table-driven.

2.2.4 Message Transfers

The CG receives and stores ASCII-string messages from other nodes on the LCN. It also is involved in the routing of host-processor messages to process operators at Universal Stations.

2.2.5 CG Database Components

The CG database can contain three types of information used by the host processor:

- Advanced Control Interface Data Points (ACIDP)
- Calculated Results Data Points (CRDP)
- Data Definition Table Internal Data Tables (Internal DDT)

Each ACIDP can have three purposes: It holds information necessary to CG scheduling of an associated ACP in the host processor; it can hold messages in transit to the ACP from other LCN nodes; it can hold calculated data values (in Custom Data Segments, see the *Control Language Reference Manual* for details) stored by the host processor (or other LCN nodes). A CG can hold no more than 250 ACIDPs.

Each CRDP holds calculated data values (in Custom Data Segments) stored by the host processor (or other LCN nodes). There is no specified connection between a CRDP and any ACP. A CG can hold no more than 500 CRDPs.

Each Internal DDT holds a list of point internal identifiers to be used by table-driven get-data requests. A CG can hold up-to-40 Internal DDTs.

See paragraph 6.2 for additional information on CG memory requirements.

2.3 SPECIAL SITUATIONS

The following paragraphs contain some additional material about CG characteristics that may be useful, depending upon the specific way in which you wish to implement your host-processor software.

2.3.1 Host Processor Scheduling of ACPs

The CG is designed to have the primary responsibility for scheduling of ACPs in the host processor. The host processor itself can schedule operation of its ACPs, but restrictions exist.

Full access to all CG functions requires an ACP to be connected to an ACIDP in the CG. If the ACP is not attached to an ACIDP, it cannot store data to CRDPs or to other LCN-resident data points, and it cannot receive messages from devices on the LCN or send messages to Universal Stations.

See Table 2-1 for a summary of scheduling restrictions based upon ACP-to-ACIDP connection and ACIDP installation mode.

Connection of an ACP to an ACIDP implies CG scheduling of the ACP, because there is no "null" scheduling option. One method around this is to establish the ACIDP as demand-only and establish logic at the host to ignore any unwanted CG activation.

For some of the transactions (messages) initiated by the host processor that normally require use of an ACIDP name, the CG accepts a "blank" name (all ASCII spaces). See Table 2-2 for a summary of these transactions.

Table 2-1 — ACP Scheduling Capabilities

	ACP Not connected to an ACIDP	ACP Is Connected; Installation Mode is		
		NORMAL	RESTRICTED	TEST
ACP can be initiated by				
CG—Cyclic	no	yes	no	no
CG—Periodic	no	yes	no	no
Operator Demand	no	yes	yes	yes
Process Special	no	yes	yes	yes
Message Waiting at CG	no	yes	yes	yes
Host Processor	yes	yes	yes	yes

Table 2-2 — Operation Request Capabilities

	"Blank" ACIDP Name accepted	Named ACIDP is in Installation Mode			
		NML	NI	RES	TES
DATA ACCESS REQUESTS					
Single point get data (ext)	yes	yes	no	yes	yes
Single point store data (ext)	no	yes	no	yes	yes
Single point get data (int)	yes	yes	no	yes	yes
Single point store data (int)	no	yes	no	yes	yes
Table get data (DDT/IDB @ host)	yes	yes	no	yes	yes
Table get data (DDT/IDB @ CG)	yes	yes	no	yes	yes
Table store data	no	yes	no	yes	yes
Get History	yes	yes	no	yes	yes
MISCELLANEOUS ACP ACTIONS					
Convert Identifier (ext to int)	yes	yes	yes	yes	yes
Build DDT/IDB	yes	yes	yes	yes	yes
Send Message to US	no	yes	no	yes	yes
Get Message from CG	no	yes	no	yes	yes
Change ACIDP Execution State	no	yes	no	yes	yes
Change ACIDP Program Mode	no	yes	yes	yes	yes
Get ACIDP Point Status	no	yes	yes	yes	yes

Legend: NML = Normal installation (ACIDP-ACP connection)
NI = Not Installed
RES = Restricted mode
TES = Test mode

2.3.2 Use of 16-Character Point Names

This release of the CG is compatible with previous TDC releases (R230 and the R300 series of releases) that use 8-character point names. It also supports the use of long point names (up to 16 characters).

Support for long point names is provided through new versions of those message transactions between the Host Processor and the CG that use point names. Each of these new messages is identified by a transaction code 100 greater than their short point name equivalent. A summary of these message types follows:

Message name	Short Point Name Transaction Code	Long Point Name Transaction Code
Operator Message Confirmation/Timeout	1	101
Turn on ACP	2	102
Get Data Request (Internal DDT in CG)	3	103
Get Data Request (Internal DDT in host)	4	104
Store Data Request	5	105
Build Internal DDT Request	6	106
Get Data Return	7	107
Store Data Return	8	108
Get Message Request	13	113
Send Message Request	15	115
ACIDP Execution State Change Request	21	121
ACIDP Program Mode Change Request	22	122
Get ACIDP Status Request	24	124
Get ACIDP Statue Return	25	125
Get Single Value Request (ext adr)	28	128
Get Single Value Return (ext addr)	29	129
Store Single Value Request (ext addr)	35	135
Store Single Value Return (ext addr)	36	136
Store Single Value Request (int addr)	37	137
Store Single Value Return (int addr)	38	138
Get Single Value Request (int addr)	39	139
Get Single Value Return (int addr)	40	140
Convert Identifier Request	41	141
Convert Identifier Return	42	142
Internal DDT Connect or Disconnect Request	45	145
Internal DDT Connect or Disconnect Return	46	146
CG Database List Request	47	147
ACIDP/CRDP/DDT Tables List Return	48	148
Get History Request	50	150
Get History Return	51	151

NOTE

When communicating with an R400 or subsequent release LCN system, the host processor should always use the 16-character-name message transactions. If the target LCN is configured for 8-character names, space fill the final 8 bits of each name.

The CG assumes that it is communicating with a pre-R400 system until it receives a "CG Database List" message (transaction code 47 or 147) from the host processor that requests "CG Node List" information (list type 5). This causes the CG to do two things.

- The CG responds with information about the CG including the point-name length (8 or 16 characters) used on the local LCN.

- The CG changes the transaction codes used for the only "unsolicited" messages that it sends to the host processor. Thus, any subsequent "Operator Message Confirmation/Timeout" messages will use the transaction code of 101 and any "Turn on ACP" messages will use the transaction code of 102.

For each host processor request where point name length is significant, response from the CG will use the transaction code that matches with the request. For example, the response to transaction code 5 is transaction code 8 while the response to transaction code 105 is transaction code 108.

For those transactions where the response does not contain a point name, there is no long point name transaction code. For example, both transaction codes 13 and 113 receive a response of transaction code 14.

2.3.3 Internetwork Transfers

The host processor can request point data or file data from remote LCNs connected to the local LCN through a Plant Information Network (PIN).

Point data transfers are directed to a specific LCN on the network by the inclusion of a two-character "PIN node ID" field in the message. This field can be left blank when the point to be accessed is on the local LCN. Long point name requests (see heading 2.3.2) can be passed through a local LCN that is configured for short point names.

Some file transfer transactions also can be sent to remote LCNs (see heading 4.10.10.1 for a list of remote file transactions). Addressing of a remote file requires the addition of a three character prefix to the file name. This prefix is composed of the two-character PIN node ID followed by a backslash (\) character (example: N1\NET>vdir>file_id.extension). A file address without the PIN prefix specifies that the file is on the local LCN.

2.3.4 Restricted and Test Operation Modes

The ACP-to-ACIDP installation mode can be one of Normal, Not Installed, Restricted, or Test. The Restricted and Test modes are intended to provide you with the capability to inhibit ACP operation in ways that you may find useful in developing debugging aids for the host processor. The only direct restriction placed by the CG on ACIDPs in either Restricted or Test mode is to prevent cyclic or periodic scheduling (host processor, operator demand and process specials only).

The CG treats ACIDPs in either Test or Restricted mode identically; however, you may wish to enforce additional use restrictions at the host processor that differentiate between those modes. For example, in Test mode you may wish to substitute test values for the data specified by a get-data request.

Certain data access-request messages do not require an ACIDP name and will be honored regardless of ACP/ACIDP connection status. See Tables 2-1 and 2-2 for summaries of restrictions enforced by the CG depending upon ACIDP installation mode.

2.4 CG/HOST STARTUP STEPS

Startup of the CG and of the host processor requires a series of steps that begins with configuration and loading of the CG, goes through establishment of communication between host processor and CG, and ends with modification of the CG database by the host processor.

2.4.1 Preparing the CG

The first step in configuring the CG—from a TDC 3000^X Universal Station—to modify the system NCF to include the CG node. Refer to *Network Data Entry* as a starting point if this procedure is unfamiliar. As part of LCN Nodes Configuration, you will assign one or more Process Units to the CG (see *Network Form Instructions* and Appendix A of this manual for additional information).

Additional CG configuration entries are made through the CG Configuration display. To reach this display, call up the Engineering Main Menu at a Universal Station on the same LCN as the CG, then select the "Computing Module" target. This brings up the Computer Gateway Build and Configuration menu from which you select the "CG Configuration" target.

Your selection of CG configuration type is controlled by the hardware used to interface your host to the LCN. Select either "CLI-HDLC" or "CLI-BISYNC" depending on the data link protocol used. Additional CG configuration choices you will have to make depend on this first selection and include the link baud rate and a floating-point number conversion type. See Section 4 of the *CG Parameter Reference Dictionary* for information on these CG configuration entries.

NOTE

On completion of CG/PLNM configuration (or configuration change), you must demand a CG checkpoint save, then shutdown and restore the node from the saved checkpoint to enable the changes to take effect.

Point building (of ACIDPs and CRDPs) can be deferred for the moment, but must precede any connection to, or use by, an ACP. Information on the building of these points and their associated Custom Data Segments is found in Appendix E of this document.

2.4.2 Starting of CG-Host Processor Communications

Responsibility for establishing communication between the host processor and CG is left to the CG. Whenever the CG is freshly loaded it attempts to make a **Cold Restart**. If communications with the host processor are interrupted following a successful Cold Restart, the CG then attempts a **Warm Restart.**

If communication is broken at any time, the failure should be recorded at the host-processor operator terminal. The CG waits for three minutes before trying again. Some failures may require reload of the CG.

2.4.3 Restart Modes

Once the Restart message is received, the host processor is expected to take the initiative for completion of the startup. The necessary steps depend on the type of Restart.

A Cold Restart requires the host processor to prepare and download necessary elements of the CG-resident database, such as ACP-to-ACIDP connections and any Internal Data Tables. This also requires the completion of the host processor's own database, by the preparation of its ACIDP/ACP status table, and the execution of any necessary external to internal point-address conversions.

A Warm Restart requires the host processor to resolve any mismatch of databases between the CG and host processor.

Upon completion of any restart, the host processor should display a restart message and the CG can begin scheduling of ACPs.

Details of the required startup messages are found starting at paragraph 4.3. Also see Table 4-2 and Figure 4-1.

2.4.4 Initial Startup Recommendations

It is recommended that any system being brought up for the first time be configured for Real-Time Journals to be kept by the History Module and to include System Error and Maintenance Recommendations Messages. See paragraph E.3.1 for some first-time scheduling recommendations.

HOST COMPUTER SOFTWARE COMPONENTS
Section 3

This section discusses the types of support programs and communication program interfaces that can be used as the base for development of host processor-resident application programs.

This section deals with the ways in which applications interfaces to CG functions can be constructed with the host-to-CG data link messages as a foundation. It is only a guide; the exact requirements/specifications can vary from user to user. The assumption is that you will develop an environment that will support the needs of your applications programs.

Some of the major elements that you will need to consider are

- Bisynch or HDLC data link
- A communications executive
- A dispatcher to activate host-resident application programs
- ACP installation/deinstallation
- A data table builder

The balance of this section is devoted to the description of a sample system that provides an application-program development and execution environment and uses the set of CG data-link messages described by this manual. This sample system description is included to provide you with design ideas for your own system.

3.1 SAMPLE SYSTEM OVERVIEW

The sample system provides an environment that permits users to develop and execute application programs that can exchange data with nodes on the TDC 3000^X LCN. In this system, the application developer works through a set of higher level interfaces that shield him from details of the host processor to CG interface. It is composed of elements in both the host processor and the Computer Gateway (CG). The following descriptions emphasize the host processor portion of those elements.

Before delving into the components and structure of the host-processor software, we first should look at the two key elements that this structure must support; the application programs (called "Advanced Control Programs" or ACPs) and the Data Definition Tables.

3.1.1 Advanced Control Programs (ACPs)

The preparation and installation of an ACP is a multi-step operation initiated at a host processor terminal. In summary, those steps are

- Prepare ACP source file(s)
- Compile and link the program(s)
- Build any required Data Definition Tables
- Use the ACP Installer to install the ACP in the host processor and to make the ACIDP connection, if CG scheduling is desired.

After the individual application program is prepared for execution in the host-processor environment, it needs to be installed as an ACP. The installation status is recorded in the host-processor's ACP Status Table. If there is an associated ACIDP, the ACP status is kept there as well.

3.1.2 Data Definition Tables (DDTs)

Data Definition Tables are required to support the table-driven data transfers. They are built at the host processor, beginning with creation of user defined source files.

Three types of Data Definition Tables are provided, Input, Output, and History. Each DDT contains an identifier by which it is known to the ACPs, and lists of ASCII point and parameter names. For non-History DDTs, other parameters provide for specialized processing of data values. These include

- Apply Calculation—Specifies whether or not to perform a calculation on the data value. A choice of 10 algorithms, some using constants and others using current data, is offered. Applies to either Get or Store Data for Real values.

- Bad—Specifies what will happen if the value is "bad." Applies to Get Data of Real or Integer data. Possibilities include

  – Leave as is
  – Substitute indicated constant
  – Substitute indicated point parameter value

- High/Low Limits—The value (or result of the calculation if used) is compared against these limits. Applies to either Get or Store Data for Real values.

- Clamp—When a limit is exceeded, specifies whether the value is clamped at its limit or replaced by the bad value constant (-0). Applies to either Get or Store Data for Real values.

- Test—When set and the ACIDP status is Test, the live value is replaced by a specified value. Applies to Get Data for Real, Integer, ASCII, or Enumeration values.

- Source/Destination—Specifies an array location in the host processor where the Get Data value is to be stored, or a Store Data value is to be taken from. Applies to Get Data and Store Data for all value types.

3.2 HOST PROCESSOR COMPONENTS

The following paragraphs deal with the components that provide the host-processor services already noted. Figure 3.1 depicts the general structure of the software and tables as implemented in the host processor portion of the sample system.

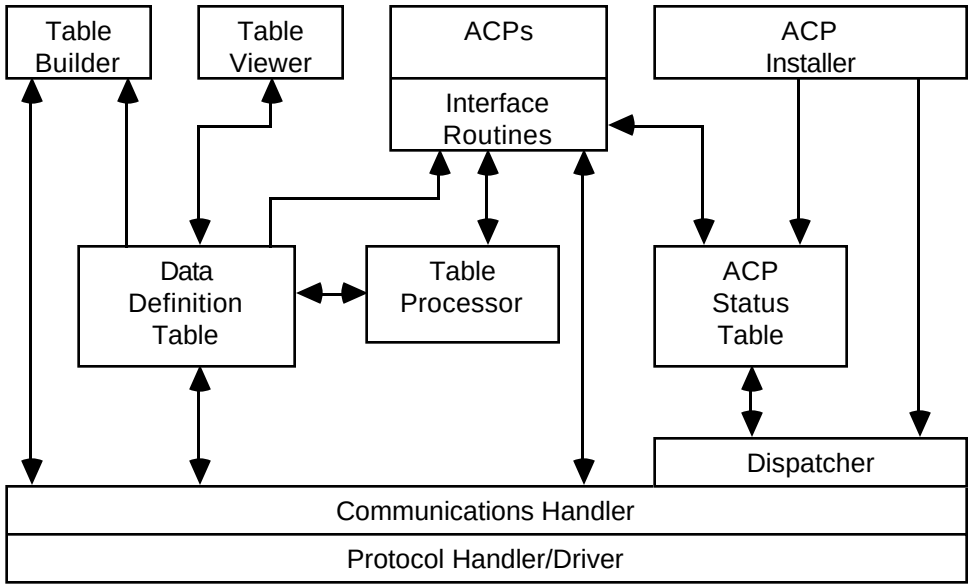


Figure 3-1— Host Processor Components of Sample System 1562

3.2.1 ACP Interface Routines

The application programmer is provided with a number of interface routines that allow him to give and get LCN data and to perform necessary housekeeping work. The specific interfaces provided are

- **Get/Store LCN Data**

  Get Data (multipoint using DDT)
  Get Single Value (external ID)
  Get Single Value (internal ID)
  Store Data (multipoint using DDT)
  Store Single Value (external ID)
  Store Single Value (internal ID)
  Get History (multipoint using DDT)
  Get Message
  Send Message

- **Scheduling and Miscellaneous functions**

  Get ACP Status
  Program Delay
  Program Termination
  ACP Trap Handler
  Check Bad Value
  Create Bad Value
  Convert External to Internal ID

3.2.1.1 Get Data

This routine fetches point-data values from the CG or elsewhere on the TDC 3000^X LCN. The specification of which data is to be fetched and where it is to be stored in the calling-program's data arrays is contained in the Data Definition Table referenced by the call.

3.2.1.2 Get Single Value (External ID)

This routine fetches data for a single-point parameter (or parameter array) from the CG or elsewhere on the LCN. The point name, parameter name, and array size if applicable, for the data to be fetched, as well as the location where it is to be stored in the host processor, are all contained in the call.

3.2.1.3 Get Single Value (Internal ID)

This routine fetches data for a single point from the CG or elsewhere on the LCN. Use of the internal point and parameter (or parameter array) address, obtained by previous use of the Convert Address function, reduces the overhead required for repetitive single-point requests. The internal identifier of the data to be fetched, as well as where it is to be stored in the host processor, are both contained in the call.

3.2.1.4 Store Data

This routine sends data to points in the CG, or elsewhere on the LCN. The specification of what points are to receive data, and the location of data within the calling-program's data arrays, is contained in the Data Definition Table referenced by the call. Errors encountered during execution of the routine, as well as individual point-data errors, are returned to the calling program.

3.2.1.5 Store Single Value (External ID)

This routine stores data to a single point in the CG, or elsewhere on the LCN. The specification of which data is to be fetched and where it is to be stored is contained in the call.

3.2.1.6 Store Single Value (Internal ID)

This routine stores data to a single point in the CG, or elsewhere on the LCN. Use of the internal point and parameter address (obtained by previous use of the Convert Address function) reduces the overhead required for repetitive single-point requests. The specification of which data is to be fetched and where it is to be stored is contained in the call.

3.2.1.7 Get History

This routine is used to fetch history data from the HM. The specification of which data is to be collected is contained in a Data Definition Table, while time bounds and the history type are calling parameters.

3.2.1.8 Get Message

This routine is used to fetch a character-string message held in a CG buffer by this program's ACIDP. The message presence is detected as the result of a Get ACP-Status request.

3.2.1.9 Send Message

This routine is used to send a message to all operator stations monitoring the unit this program's ACIDP was assigned to when configured. A request to wait for operator confirmation is optional. If operator confirmation is requested, execution of the requesting program is suspended until either the confirmation occurs or until its specified wait time expires. The requesting program receives an indication of whether confirmation or a time out occurred.

3.2.1.10 Get ACP Status

This routine fetches a set of parameters that enables the requesting program to determine why the system has turned it on, and what special processing may be required at this time.

3.2.1.11 Program Delay

This routine suspends execution of the calling ACP for a specified number of seconds. Program execution resumes at the statement following the delay call.

3.2.1.12 Program Termination

This routine terminates the execution of the calling ACP. It must be used as the last operating statement of each ACP. If there is an associated ACIDP, a termination-status code is stored in its ABORTCOD parameter. If an ACP is aborted by the operating system, a system-assigned abort code is stored.

The execution state of an ACIDP can be changed from ABORT to normal by operator demand through a Universal Station. This causes the execution state to be changed from ABORT to either OFF or DELAY. A subsequent operator demand will activate the ACP if its activation type permits.

3.2.1.13 ACP Trap Handler

This routine is required by the host processor operating system to ensure the export of an Abort Code to the CG in event of abnormal termination of the ACP.

3.2.1.14 Check Bad Value

This routine checks a value of type Real to determine if it is a valid single-precision, floating-point number. Its primary purpose is to check for the "Bad Value" indicator.

3.2.1.15 Create Bad Value

This routine stores the Bad Value constant (-0) into the specified variable of type Real.

3.2.1.16 Convert External to Internal ID

This routine converts the external point and parameter name for a single point to its internal form (through communication of a request to the CG). System overhead is reduced through use of the internal address by repetitive single-value data gets and stores.

3.2.2 ACP Installer

The ACP Installer is an interactive program that allows you to

- Install (or remove) ACPs in the host processor
- Connect (or disconnect) ACPs to ACIDPs in the CG
- Connect (or disconnect) CG-resident Internal DDTs to ACIDPs in the CG
- Activate an ACP (if ACIDP connection exists, only if installation mode is Test or Restricted)
- Change an ACP program's mode (Normal, Restricted, Test, Not Installed)
- Examine the ACP Status Table
- Examine past ACP Installer Activity
- Set relative priority level of ACP
- Make ACP permanently memory-resident

After an ACP has been compiled and linked, this function is used to make the ACP known to the host processor by adding its pathname to the ACP Status table. If the ACP has an associated ACIDP, the installer sends the pathname to the ACIDP. This name is used by both the host processor and the CG as the method for identifying which ACP is to be affected.

The ACP Installer also is used to change an ACP's installation mode (among Normal, Restricted, and Test) or to uninstall an ACP (both at the host processor and at the CG). It is used to activate those ACPs without an ACIDP and those whose ACIDP is in Restricted or Test state. The ACP Installer also provides commands to list or print the ACP status table.

Any required ACIDPs can be built either before or after ACP compilation and linking; however, they must be built before the ACP-to-ACIDP connection can be made.

3.2.3 ACP Status Table

The host processor's ACP Status table contains the following information on each installed ACP:

- ACP name (and Task Group when activated)
- ACIDP name (if connection established)
- ACP installation mode (Not Installed, Normal, Test, Restricted)
- ACP execution state (Off, Run, Abort, Fail, Accessing)
- ACP Turn-on information

When its installation mode is Restricted or Test, any attempted parameter store by the ACP to the LCN is stopped. The blocked data can be viewed by using the Data Table Viewer program.

When its installation mode is Restricted or Test, an ACP will not be activated by periodic or cyclic scheduling, but can be activated by operator demand, a Point Process Special, or through the host processor terminal (through the Installer).

When its installation mode is Test, the data values returned to an ACP by a table-driven data-access request are replaced by test values contained in the referenced Data Definition Table.

During the time that an ACIDP is Inhibited by Operator action at a Universal Station, the ACP will not be activated by the CG either by schedule or by PPS.

3.2.4 Data Definition Table Builder

The DDTs are prepared and stored separately from ACPs. The ACP specifies a DDT in the data access fetch/store request. An ACP can use one or more tables (only one at a time), but use of a DDT by more than one ACP can create file access problems in the host processor.

The Definition Table Builder (DTB) creates five files for each DDT:

- External Point Names—This is separate from the user source file, and is used by the Data Table Builder in rebuild operations.
- Internal Data Tables (Internal DDT)—These are the lists of converted point names that are used to identify the point-parameter values to be accessed. Selection of which input Internal DDTs are to reside in the CG is a Table Builder function.
- Transform File—This file contains all the user-defined values used in limit checking, clamping, conversion calculations, data substitution, etc.
- Returned Value Status—This file contains the latest returned or stored data values; both before and after conversion calculations values are included.
- Table Attributes—This file contains current status information on the DDT, including the number of points of each data type it contains.

3.2.5 Definition Table Processor

The Definition Table Processor (DTP) acts only on current data requests (not on History). For a current value-data fetch, the DTP processes the input value according to the rules and destination specified in the DDT, then places the data in a table area specified by the ACP. For a data store, the ACP output-request values are processed according to the rules specified by the table parameters, then the processed data is sent to the CG. The DTP optionally saves the values, both raw and processed, and value status in the Value/Status table of the DDT.

3.2.6 Definition Table Viewer

The Definition Table Viewer (DTV) provides the engineer with the means of examining a DDT that has been used by an ACP and processed by the DTP. The DTV is invoked and values are displayed at a host-processor terminal. The parameters and values are gathered into a temporary file that can be displayed or printed. Selected DDT entries such as limits or substitution algorithms can be changed through the DTV (the DDT source file is unchanged).

3.2.7 Dispatcher

The Dispatcher activates ACPs when requested by the Communications Handler or by the ACP Installer.

3.2.8 Communications Handler

The Communications Handler is responsible for "blocking" and "unblocking" messages between the CG and the host processor. Additionally, it is responsible for message routing over the two links (if a second link is present).

The Communications Handler receives messages from host processor programs, breaking them into smaller size transmission blocks as necessary, adds identification, and sends the packets to the CG through the Protocol Handler/Driver.

The Communications Handler receives the message packets from the CG, assembles the blocks to reform the message, and routes the message to the program (Task Group ID) specified in the message.

DATA LINK MESSAGES
Section 4

This section covers the various data transfer and support messages that pass between the CG and host processor.

4.1 MESSAGE STRUCTURE AND FORMATS

In this section, emphasis is placed on the messages as they may be presented by you to the applications program. For example, the physical link messages must be chopped into 512-word or smaller transmission blocks, but this should be kept invisible to the application user who can work with messages of up to 4000 data words for single-point transfers and up to 5200 data words for multipoint transfers. Data link operation and the format of messages as passed over the link are discussed in Section 5.

A 16-bit data word length is presumed.

4.1.1 Message Sequences

Communications between the Host Processor and the CG have to be considered on three levels:

- **Driver level**—This level is concerned with link control, including block retransmission when necessary.
- **Communication Handler level**—At this level, the longer messages are broken into blocks for transmission, and are reconstituted on reception. Whole-message checking and retransmission is performed. Confirms that the link is in operation and performs failback from dual-link to single-link operation when required.
- **Applications level** —Messages generated at this level are concerned with the exchange of data between host and CG.

Driver-level requirements are explained in Section 5, while the content and uses of messages exchanged at the Communications Handler and Applications levels are explained in this section.

The messages exchanged at the Communications Handler level are summarized in Table 4-1. Details of their operation start at paragraph 4.2.

Messages at the Applications Level fall into two categories: normal operation sequences and initialization sequences.

- Initialization message sequences are initiated by the CG in response to either of these events:
  - CG power up and software load (cold restart)
  - Data link-communications restart (warm restart)

Table 4-2 shows the set of startup messages, while Figure 4-1 illustrates a simple startup-message sequence including components at the communications level. Details of the initialization messages are found at paragraph 4.3.

- Normal Operation-message sequences are initiated by the host processor, each with a required CG response message. These request-response pairings are shown in Table 4-3, and an example of a normal operation-message exchange, including components at the communications level, is shown in Figure 4-2.

Table 4-1 — Communications Handler Level Messages

MESSAGE TYPE	TRANSACTION CODE	SENT BY	NOTES
Time Synch	31	CG	Requires host to echo back same message
Confirmation	30	either	Required response to any applications-level message. Negative confirmation requires sender to retransmit the message.
Link Fail	32	either	Used to initiate failover from dual-link to single-link operation

Table 4-2 — Applications Level Messages During Initialization or Restart

MESSAGE TYPE	TRANSACTION CODE	SENT BY	REQUIRED RESPONSE
CG Restore	43	CG	Zero or more of the following request types followed by Restore Complete (44)
DB List Request	47/147	Host	Database List Response (48/148)
ACIDP Prog Mode Chg	22/122	Host	Program Mode Change Response (23)
ACIDP Exec State Chg	21/121	Host	none
Store Internal DDT in CG	10	Host	Store Internal DDT response (12)
Delete Internal DDT from CG	11	Host	Delete Internal DDT response (12)
Internal DDT Con/Discon	45/145	Host	Connect/Disconnect response (46/146)
Restore Complete	44	Host	none

Table 4-3 — Applications Level Messages During Normal Operation

MESSAGES CATEGORIES	HOST REQUEST TRANSACTION CODE	CG RESPONSE TRANSACTION CODE	NOTES
PROGRAM SCHEDULING AND STATUS MESSAGES			
ACIDP Execution St Chg	21/121	none	
ACIDP Program Mode Chg	22/122	23	
Get ACIDP Status	24/124	25/125	
DDT Connect/Disconnect	45/145	46/146	
Turn On ACP	--	2/102	-unsolicited
INTERNAL DDT PREPARATION MESSAGES			
Build Internal DDT	6/106	9	
Store Internal DDT in CG	10	12	
Delete Internal DDT from CG	11	12	
POINT-DATA TRANSFERS USING DATA TABLES			
Get Data	3/103	7/107	-Internal DDT in CG
Get Data	4/104	7/107	-Internal DDT in host
Store Data	5/105	8/108	
SINGLE-POINT DATA TRANSFERS			
Get Single Point	28/128	29/129	-external identifier
Store Single Point	35/135	36/136	-external identifier
Convert Identifier	41/141	42/142	
Get Single Point	39/139	40/140	-internal identifier
Store Single Point	37/137	38/138	-internal identifier
HISTORY DATA TRANSFERS			
Get History	50/150	51/151	
TEXT MESSAGE TRANSFERS			
Get Message	13/113	14	
Send Message	15/115	16	-immediate response
		1/101	-confirmation (if req'd)
FILE TRANSFERS			
File Request/Response	62	63	

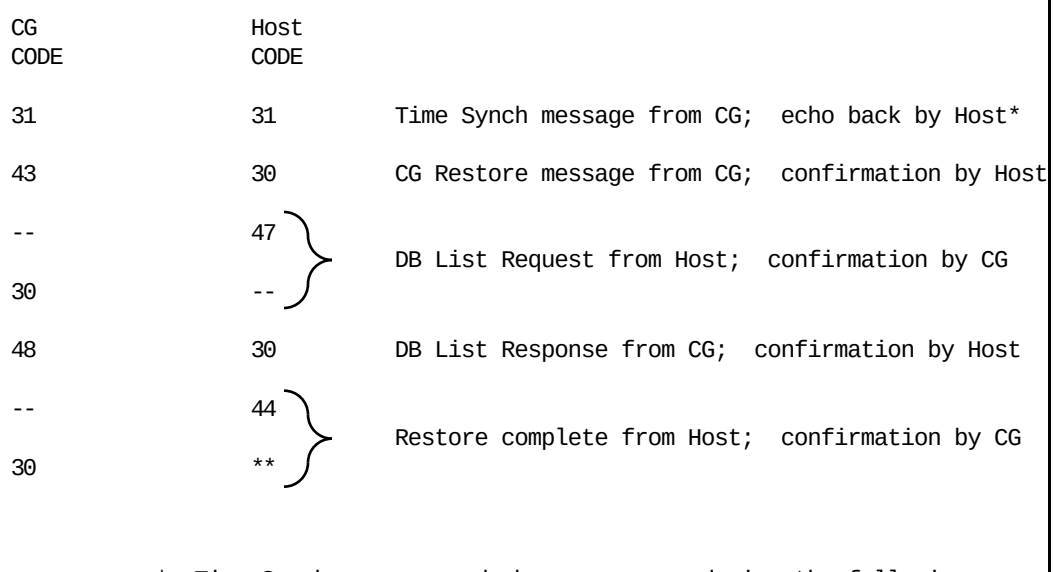


Figure 4-1— Example of Startup Messages Sequence

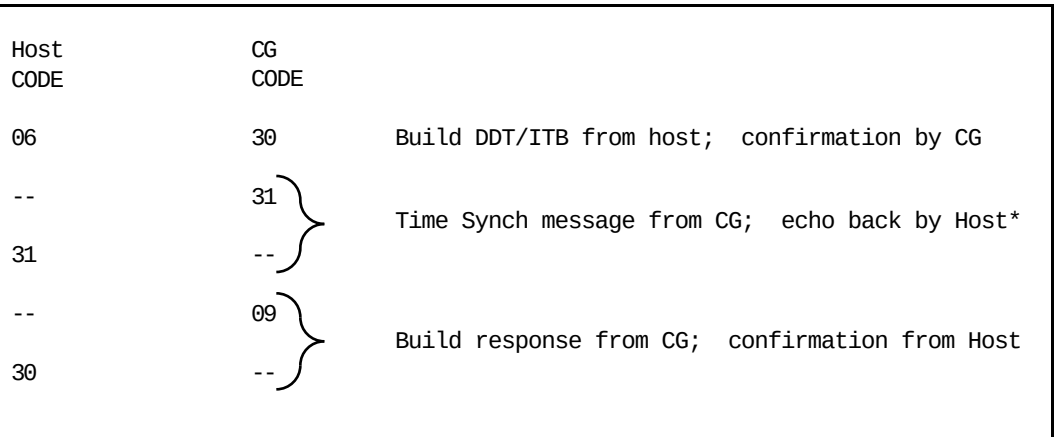


Figure 4-2 — Example of Normal Operation Messages Sequence

4.1.2 Data Link Messages Format

Messages passed between the CG and host processor are formed of a fixed-length message header followed by message data that varies in length by message type and by content.

4.1.2.1 Message Header Format

The header for each message contains eight words. Header words 3, 4, 7, and 8 are used only at the link-operation level. The remaining words are used for internal routing and processing. Significance of each header word is explained below.

Header word 1 Transaction Code (integer)—A unique code number is assigned to each message type. Table 4-4 shows all currently assigned Transaction Code values, along with additional message information.

Header word 2 Number of Words (integer)—The meaning assigned to this word can vary depending on whether you are at the applications level or at the communications-handler level. At the applications level, this is the number of information words in the entire message (excludes the 8-word message header). At the link level, this is the number of information words contained within a single transmission block (the message header appears in each transmission block, but is not included in this count).

Header word 3 Number of Blocks (integer)—This is the total number of blocks into which this message is subdivided for transmission.

Header word 4 Block Number (integer)—A counter sequentially assigned to each transmission block of a message.

Header word 5 CG Identifier (integer)—This identifies specific processing functions in the CG.

1=Scheduler
2=Data Access Handler
3=Miscellaneous Functions Handler
5=Communications Handler

Each message is assigned one of these identifiers (see Table 4-1). If the Transaction Code and the CG Identifier are not properly matched, the CG will halt on an error.

Header word 6 Host Task Identifier (2 ASCII characters)—This field has fixed values for three CG-initiated messages and one host-initiated message.

- Turn on ACP (Transaction Code 2/102)—"DI"
- Time Synch (Transaction Code 31)—"UN"
- CG Restore (Transaction Code 43)—"RS"
- Restore Complete (Transaction Code 44)—"RS"

For all other message types, the codes are defined by the host processor and are echoed in the CG responses. The Host Task Identifier is intended as a message-routing code to be used by the host processor.

Header word 7 Message Tag (integer)—For internal CG use only.

Header word 8 Message Counter (integer)—For a dual-Bisynch link, this is a counter (1..30000) sequentially assigned to each Data message (not to Confirmation or Time Synchronization messages). The "send" and "receive" counts are separate. After message 30000, the counter resets to one. For a single link this value is not checked by the CG.

Note that header word 8 has a different meaning in negative Confirmation Messages. See Paragraph 4.2.2 for details.

4.1.2.2 Message Data Formats

The format and content of data in each message is covered in the discussion of the individual messages beginning at paragraph 4.2

4.1.2.3 Error Fields in CG Return Messages

Most Applications-Level data link messages are paired into requests and returns (see Table 4-3). Return messages from the CG can contain two types of error information:

- The Return Status word (found in all returns) shows whether or not the request was successfully processed, and if not, what type of error was involved.
- Value Status words (found in data get or store returns) show if there have been any data access errors that would invalidate the requested operation for a specified point-parameter. There are over 200 different data access error codes that can be returned (see the *Messages Directory* for a complete list of Data Access Status Codes).

Table 4-4 — CG Sources and Destinations of Data Link Messages

CG SCHEDULER (Identifier parameter=1)

TRANSACTION CODE	MESSAGE NAME	WHERE DESCRIBED	MESSAGE INITIATOR
1/101	Operator Message Confirmation/Timeout	4.9.5	CG
2/102	Turn On ACP	4.4.5	CG
31	Time Synchronization	4.2.1	CG*
43	CG Restore	4.3.1	CG
44	Restore Complete	4.3.2	host

CG DATA ACCESS HANDLER (Identifier parameter=2)

TRANSACTION CODE	MESSAGE NAME	WHERE DESCRIBED	MESSAGE INITIATOR
3/103	Get Data Request (Internal DDT in CG)	4.6.1	host
4/104	Get Data Request (Internal DDT in host)	4.6.2	host
5/105	Store Data Request	4.6.4	host
6/106	Build Internal DDT Request	4.5.1	host
7/107	Get Data Return	4.6.3	CG
8/108	Store Data Return	4.6.5	CG
9	Build Internal DDT Return	4.5.2	CG
28/128	Get Single Point Request (ext addr)	4.7.2	host
29/129	Get Single Point Return (ext addr)	4.7.4	CG
35/135	Store Single Point Request (ext addr)	4.7.5	host
36/136	Store Single Point Return (ext addr)	4.7.6	CG
37/137	Store Single Point Request (int addr)	4.7.11	host
38/138	Store Single Point Return (int addr)	4.7.12	CG
39/139	Get Single Point Request (int addr)	4.7.9	host
40/140	Get Single Point Return (int addr)	4.7.10	CG
41/141	Convert Identifier Request	4.7.7	host
42/142	Convert Identifier Return	4.7.8	CG
50/150	Get History Request	4.8.1	host
51/151	Get History Return	4.8.2	CG
62	File Transfer Request	4.10	host
63	File Transfer Response	4.10	CG

* Each Time Synchronization message sent by the host to CG, is in response to a Time Synchronization message initiated by the CG.

(Continued)

Table 4-4 — CG Sources and Destinations of Data Link Messages (continued)

CG MISC FUNCTIONS HANDLER (Identifier parameter=3)			
TRANSACTION CODE	**MESSAGE NAME**	**WHERE DESCRIBED**	**MESSAGE INITIATOR**
10	Store Internal DDT in CG Request	4.5.3	host
11	Delete Internal DDT from CG Request	4.5.4	host
12	Internal DDT Store/Delete Return	4.5.5	CG
13/113	Get Message Request	4.9.1	host
14	Get Message Return	4.9.2	CG
15/115	Send Message Request	4.9.3	host
16	Send Message Return-Immediate	4.9.4	CG
21/121	ACIDP Execution State Change Request	4.4.6	host
22/122	ACIDP Program Mode Change Request	4.4.1	host
23	ACIDP Mode Change Return	4.4.2	CG
24/124	Get ACIDP Status Request	4.4.3	host
25/125	Get ACIDP Status Return	4.4.4	CG
45/145	Internal DDT Connect or Disc. Request	4.4.7	host
46/146	Internal DDT Connect or Disc. Return	4.4.8	CG
47/147	CG Database List Request	4.3.3	host
48/148	ACIDP List Return	4.3.4	CG
48/148	CRDP List Return	4.3.5	CG
48/148	Internal DDT Tables List Return	4.3.6	CG
48/148	Enhanced ACIDP List Return	4.3.7	CG

CG COMMUNICATIONS HANDLER (Identifier parameter=5)			
TRANSACTION CODE	**MESSAGE NAME**	**WHERE DESCRIBED**	**MESSAGE INITIATOR**
30	Message Confirmation	4.2.2	either
32	Data Link Failure	4.2.3	either

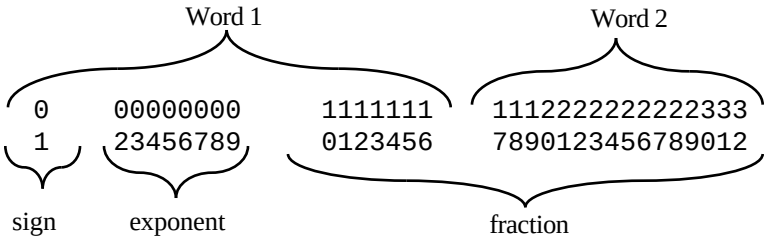
4.1.3 Data Representations

NOTE

In any data-exchange message (except for History) you are required to specify the data type for each data-point parameter being transferred. You also will need to know the set of value assignments (both ordinal and character string) for any enumerations accessed.

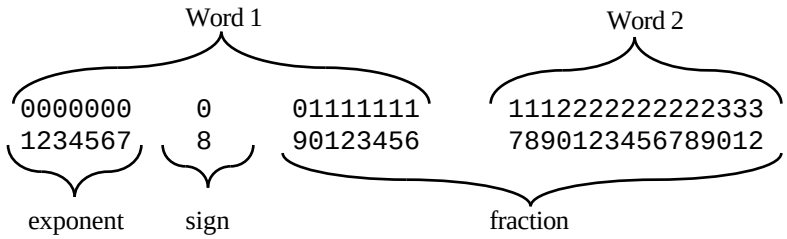
REALS—Real values consist of 32 bits. The format seen at the host processor depends on configuration choice and will be one of the following:

IEEE 754

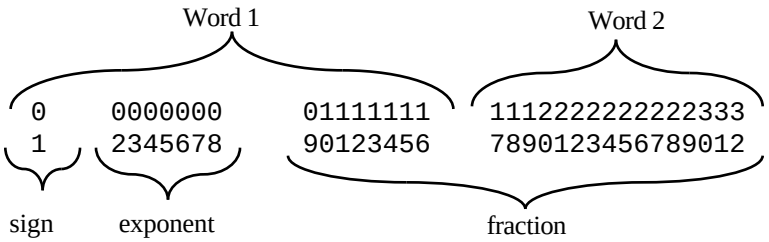


In the IEEE 754 format, the exponent is represented by 2^E where E = value in the exponent field -127. The fraction is left justified such that a hidden 1 bit exists. A positive or negative value with an exponent of 255 and a non-zero fraction is used to indicate a Bad Value (also referred to as Not a Number or NaN). TDC 3000^X most commonly generates the hexadecimal value 7F800001 to represent a Bad Value.

DPS6



IBM 370



In the DPS6 and IBM formats, the exponent is represented by 16^E where E = value in the exponent field -64. The fraction is composed of six binary coded hexadecimal digits, normalized so that the first digit is not zero. The form -0 (sign bit only set) is used to indicate a Bad Value, also referred to as Not a Number (NaN).

The CG handles necessary conversions to/from the DPS6 or IBM format from/to the format used by devices on the LCN (IEEE 754).

INTEGER—Each integer value occupies one 16-bit word with a value range of -32767 to 32767. History messages also include 32-bit integer values for sample counts.

ASCII—Each ASCII value consists of a 24- or 40-byte ASCII string. A string of question marks is used to specify a "doubtful" ASCII value.

ENUMERATION—There are two ways to represent enumerated values; as strings of eight ASCII characters or as ordinal (integer) values. The choice of representation is made within the various get/store data messages.

TIME—The CG has access to three representations of LCN time.

- LCN external time consists of an 18-byte ASCII string whose format is MM/DD/YYΔHH:MM:SSΔ (where "Δ" represents a space).
- LCN internal time is contained in three 16-bit words. The first two words form a 32-bit unsigned integer that contains a count in seconds since 00:00:00 on January 1, 1979. The third word is an integer value that represents the count of tenths of milliseconds in the current second.
- Timestamps returned with history data consist of 32-bit unsigned integers that contain the count in seconds since 00:00:00 on January 1, 1979.

EXTERNAL ENTITY ID—(Read Only) Consists of an up-to-16-character ASCII string that names an LCN-resident entity (used for CL program dynamic indirection). This is followed by two ASCII characters that identify the LCN.

INTERNAL ENTITY ID—Consists of four 16-bit words containing the internal identification of an LCN-resident entity (used for CL program dynamic indirection) as follows:

Word 1 - Byte 1 - entity type
- Byte 2 - unit association

Word 2 - Byte 1 - LCN number (0 = local)
- Byte 2 - function set

Word 3 - Byte 1 - data realm
- Byte 2 - revision number

Word 4 - local routing code

4.1.4 Field-Specific Information

POINT NAME—The point name must be from one-to-eight characters long and begin with "A..Z," "0..9," or "$." Characters within the point name must be "A..Z," "0..9," or "_" (underscore). Consecutive underscore characters, or trailing underscores are not permitted. The first trailing "blank" character signals end of the name.

POINT NAME INDEX—Normally will equal zero. This field is not used with user-defined values; it applies only to certain system data-base points.

PARAMETER—The parameter name must be from one-to-eight characters long and begin with "A..Z." Characters within the parameter must be "A..Z," "0..9," or "_" (underscore). Consecutive underscore characters, or trailing underscores are not permitted. The first trailing "blank" character signals end of the name. See the *Computer Gateway Parameter Reference Dictionary, Hiway Gateway Parameter Reference Dictionary* and *Network Gateway Parameter Dictionary* for information about TDC 3000^X standard parameters.

PARAMETER INDEX—Applies only to certain parameter types (see *Computer Gateway Parameter Reference Dictionary and Hiway Gateway Parameter Reference Dictionary*); when not applicable must equal zero.

INTERNAL DDT TABLE NAME—The Internal DDT Table name must be from one-to-nine printing ASCII characters long and contain no leading or embedded blanks.

ACP NAME—The ACP name as stored in the ACIDP parameter ACPROG when the ACP is attached. It must be from one-to-twelve printing ASCII characters long and contain no leading or embedded blanks.

4.1.5 Data Definition Tables

The most efficient method for acquiring data from, or sending data to, other LCN nodes is by means of prebuilt Data Definition Tables (DDTs) that contain all necessary point specific data (including the internal data point addresses).

These tables are used for three types of data transfers: input or output of current values and input of history. The allowed data types and maximum point counts for each type are shown in Table 4-5, below.

Table 4-5 — Data Definition Table Types

TABLE TYPES	ALLOWED DATA TYPES	PT COUNT MAX
Input	Real	300
	Integer	300
	ASCII (24- or 40-Character)	152
	Enumeration	300
	Internal Entity id	300
	External Entity id	300
	LCN Internal Time	300
Output	Real	300
	Integer	300
	ASCII (24- or 40-Character)	152
	Enumeration	300
	Internal Entity id	300
	LCN Internal Time	300
History	See heading 4.8	24

Two forms of data tables are required. The External (source) DDT consists of a list of external data-point names (point name and parameter), and is sent by the host processor to the CG. The Internal DDT contains internal point addresses used for LCN data access, and is returned by the CG to the host processor. For current values (non-history) input only, a copy of the internal table can be held by the CG as a way to reduce data-link overhead when processing data-input requests.

Note that these Internal DDTs need to be rebuilt any time that any referenced point in the LCN database is deleted and/or reloaded.

4.2 COMMUNICATIONS HANDLER MESSAGES

The messages in this group are designed for use at the communication-handler level and are not intended to be visible at the applications level.

4.2.1 Time Synchronization Message

This unsolicited message from the CG sends current time and date to the host processor and is used by the CG during periods of host-processor inactivity to confirm that the data link and host processor are ready to restart or continue to function properly. This is the first message sent by the CG to begin the link-restart process.

Header information: Word 1—Transaction Code = 31
Word 2—Number of Words = 11
Word 5—CG Identifier = 1
Word 6—Host Task Identifier = UN

Message format: Words 1..11—22-character ASCII string containing date and time in format MM/DD/YYbHH:MM:SS:NNNN (NNNN is a millisecond counter).

Use information: The Time Synchronization Message does not require a Message Confirmation. Instead, it requires an echo back of this message by the host.

The host processor must echo back a Time Synchronization message before the CG can send a Restart message. During restart or during normal operation, the host processor must echo a Time Synchronization message (or initiate any Data Message or Confirm receipt of a message) within 1000 seconds to maintain communication.

Time Synchronization messages are sent at 5-second intervals until restart has begun, then at 15-second intervals at a priority lower than that of any data messages.

4.2.2 Message Confirmation

This message is used to confirm that all blocks of a message have been properly received. Each message except Time Synchronization or Link Failure (and the Message Confirmation message itself) must be confirmed by the receiver (either host processor or CG) before the next message can be sent by the originator. A negative confirmation requires the originator to retransmit all blocks of the original message.

Header information:

Word 1—Transaction Code = 30
Word 2—Number of Words = 1
Word 5—CG Identifier = 5
Word 8—Reason for Negative Confirmation
0=Block out of sequence
1=Illegal Transaction Code
2=Illegal CG Identifier
3=Number of words out of range
4=Number of blocks out of range
5=Block data too small
6=Block data size conflict
7=Block header conflict
8=Message size conflict
9=Message timeout

Message format:

Word 1—Confirmation status (integer) 1=negative
2=positive

Use information:

Note the use of header word 8 in a negative message confirmation to indicate the CG's reason for rejecting the message. These error reasons generally point to software problems at the host processor.

For single-block messages, the Number of Blocks and Block Number must both equal one.

For multi-block messages, the Block Number of each succeeding block must increase sequentially from one to the Number of Blocks value.

Any problems with format or content of Time Synchronization or Message Confirmation messages as received by the CG (or timeout on either) causes the CG to immediately stop link communications and to issue a System Error Journal report.

With each negative Message Confirmation message, the CG also issues a report to be included in the LCN System Error Journal. See paragraph 5.7 for an illustration of these reports.

4.2.3 Link Failure Message

This message is sent by either CG or host processor to indicate failure of one of the links in a dual-Bisynch pair. It is used as a trigger for switch-over to single-link communication.

Header information: Word 1—Transaction Code = 32
Word 2—Number of Words = 1
Word 5—CG Identifier = 5

Message format: Word 1—not used

Use information: On detection of a failure of its input link, the CG or the host sends this message on its output link, immediately followed by EOT. Both processors then revert to single-link communication (requiring a line bid before each transmission).

4.3 DATA LINK INITIALIZATION MESSAGES

The CG will not start the scheduling of ACPs or perform any of the normal LCN data exchanges until it has exchanged a set of initialization messages with the host processor. The CG begins by sending a time-synchronization message (see paragraph 4.2.1). When the time-synchronization message is echoed back from the host processor, the CG sends a message that informs the host processor of the type of restart.

For "cold" restarts, the host processor must restore the CG's database by sending down any CG-resident Internal DDTs and reconnecting ACPs to their ACIDPs. It then must send a message indicating that all initialization procedures are complete. For "warm" restarts, the host processor must ensure agreement between its database and that of the CG before indicating that the initialization is complete. See paragraph 5.5 for additional information on the message sequences involved in moving from initialize state to normal operation.

4.3.1 CG Restore

This message originates at the CG and informs the host processor that the link is to be restarted. Message content informs the host processor of the restart type, which is either

Cold Restart—The CG has been reloaded (initial startup or recovery from a CG failure)

Warm Restart—The CG has not been reloaded (recovery from host processor or data link failure)

Header information:

Word 1—Transaction Code = 43
Word 2—Number of Words = 2
Word 5—CG Identifier = 1
Word 6—Host Task Identifier = RS

Message format:

Word 1—Restart type (integer) 1=cold restart
2=warm restart

Word 2—Startup-message number (integer) that is compared with the returned message in case of multiple restart attempts.

Use information:

If the CG has been reloaded, the host processor must retransmit any data necessary to bring the CG database current with that in the host processor.

Note that the CG may be reloaded from a current checkpointed file and, therefore, may already match up with the host computer's database.

4.3.2 Restore Complete

This message is returned by the host processor in response to the CG Restore message, but only after completion of its initialization/synchronization process. Message content must be identical to that in the original restore message.

Header information:

Word 1—Transaction Code = 44
Word 2—Number of Words = 2
Word 5—CG Identifier = 1
Word 6—Host Task Identifier = RS

Message format:

Word 1—Restart type (integer) 1=cold restart
2=warm restart

Word 2—Startup message number (integer) that is taken from the restore message being responded to.

Use information:

Until this message has been sent and acknowledged, the host should send only the following application-level message types:

- Transaction 47/147—CG Database List request
- Transaction 10—Store Internal DDT in CG request
- Transaction 11—Delete Internal DDT from CG request
- Transaction 45/145—Internal DDT Connect or Disconnect request
- Transaction 21/121—ACIDP Execution State Change request
- Transaction 22/122—ACIDP Program Mode Change request

There is a 3-minute window from dispatch of the Restore message by the CG for return of the Restore Complete. If this limit is exceeded, the CG sends another Restore message. Any tardy response to the original Restore message is ignored by the CG.

4.3.3 CG Database List Request

This message is originated by the host processor to determine content of the CG database. Each request specifies one of five list types: an ACIDP list, an enhanced ACIDP list, a CRDP list, a Internal DDT list, or a CG node list.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 47
Word 2—Number of Words = 1
Word 5—CG Identifier = 3

Message format: Word 1—List type (integer) 1=ACIDP list
2=CRDP list
3=DDT list
4=enhanced ACIDP list
5=CG node list

Use information: This request is used during startup (or resumption) of communications between the host processor and the CG as part of the process of ensuring database agreement. See subsection 2.3.2 for important information about other effects of this transaction when a CG node list is requested.

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 147
Word 2—Number of Words = 1
Word 5—CG Identifier = 3

Message format: Word 1—List type (integer) 1=ACIDP list
2=CRDP list
3=DDT list
4=enhanced ACIDP list
5=CG node list

Use information: This request is used during startup (or resumption) of communications between the host processor and the CG as part of the process of ensuring database agreement. See subsection 2.3.2 for important information about other effects of this transaction when a CG node list is requested.

4.3.4 ACIDP List Return

This message is returned by the CG if the host processor has requested the ACIDP list.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 48
Word 2—Number of Words = 2 plus 12 times the number of ACIDPs
Word 5—CG Identifier = 3

Message format:
Word 1—Return status (integer) 0=normal status
1=invalid type
Word 2—Number of ACIDPs (integer)
Word 3—start of ACIDP list

ACIDP list format
Words 1..4—ACIDP name (8 ASCII characters)
Words 5..10—associated ACP name (12 ASCII characters)
Word 11—program status (integer)
0=NOT INST
1=TEST
2=RESTRICT
3=NORMAL
Word 12—execution state (integer)
0=ABORT
1=ACCESS
2=DELAY
3=OFF
4=RUN
5=WAIT
6=FAIL

Use information:
There can be a maximum of 250 ACIDPs in a CG.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 148
Word 2—Number of Words = 2 plus 16 times the number of ACIDPs
Word 5—CG Identifier = 3

Message format:
Word 1—Return status (integer) 0=normal status
1=invalid type
Word 2—Number of ACIDPs (integer)
Word 3—start of ACIDP list

ACIDP list format

Words 1..8—ACIDP name (16 ASCII characters)
Words 9..14—associated ACP name (12 ASCII characters)
Word 15—program status (integer)
0=NOT INST
1=TEST
2=RESTRICT
3=NORMAL
Word 16—execution state (integer)
0=ABORT
1=ACCESS
2=DELAY
3=OFF
4=RUN
5=WAIT
6=FAIL

Use information:

There can be a maximum of 250 ACIDPs in a CG.

4.3.5 CRDP List Return

This message is returned by the CG if the host processor has requested the CRDP list.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 48
Word 2—Number of Words = 2 plus 4 times the number of CRDPs
Word 5—CG Identifier = 3

Message format Word 1—Return status (integer) 0=normal status
1=invalid type
Word 2—Number of CRDPs (integer)
Words 3..6—Name of CRDP number 1 (8 ASCII characters)
Words 7..10—Name of CRDP number 2 (8 ASCII characters)
Words nn..nn+3—Name of CRDP number N (8 ASCII characters, nn=4N-1)

Use information: There can be a maximum of 500 CRDPs per CG.

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 148
Word 2—Number of Words = 2 plus 8 times the number of CRDPs
Word 5—CG Identifier = 3

Message format Word 1—Return status (integer) 0=normal status
1=invalid type

Word 2—Number of CRDPs (integer)
Words 3..10—Name of CRDP number 1 (16 ASCII characters)
Words 11..18—Name of CRDP number 2 (16 ASCII characters)
Words nn..nn+3—Name of CRDP number N (16 ASCII characters, nn=8N-5)

Use information: There can be a maximum of 500 CRDPs per CG.

4.3.6 DDT List Return

This message is returned by the CG if the host processor has requested the DDT list.

Eight-Character Tagname Form

Header information:
Word 1—Transaction-Code = 48
Word 2—Number of Words = 2 plus 5 times the number of Internal DDT tables in-CG
Word 5—CG Identifier = 3

Message format:
Word 1—Return status (integer) 0=normal status
1=invalid type
Word 2—Number of Internal DDTs (integer)
Words 3..7—Internal DDT table name 1 (9 ASCII characters)
Words 8..12—Internal DDT table name 2 (9 ASCII characters)
Words nn..nn+4—Internal DDT table name N (9 ASCII characters, nn=5N-2)

Use information:
There can be a maximum of 40 Internal DDTs per-CG.

Sixteen-Character Tagname Form

Except for the Transaction Code (148), the sixteen-character tagname form for this message is identical to the eight-character tagname form.

4.3.7 Enhanced ACIDP List Return

This message is returned by the CG if the host processor has requested the enhanced ACIDP list.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 48
Word 2—Number of Words = 2 plus 20 times the number of ACIDPs
Word 5—CG Identifier = 3

Message format:
Word 1—Return status (integer) 0=normal status
1=invalid type
Word 2—Number of ACIDPs (integer)
Word 3—start of ACIDP list

ACIDP list format
Words 1..4—ACIDP name (8 ASCII characters)
Words 5..10—associated ACP name (12 ASCII characters)
Word 11—program status (integer)
0=NOT INST
1=TEST
2=RESTRICT
3=NORMAL
Word 12—execution state (integer)
0=ABORT
1=ACCESS
2=DELAY
3=OFF
4=RUN
5=WAIT
6=FAIL
Words 13..17—precollect DDT name (9 char, lsb unused)
Word 18—SCH trigger (integer) 0=OFF 1=ON
Word 19—PPS trigger (integer) 0=OFF 1=ON
Word 20—DMD trigger (integer) 0=OFF 1=ON

Use information:
There can be a maximum of 250 ACIDPs in a CG.

The information contained in list format words 13-20 is not available for display at a Universal Station.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 148
Word 2—Number of Words = 2 plus 24 times the number of ACIDPs
Word 5—CG Identifier = 3

Message format:
Word 1—Return status (integer) 0=normal status
1=invalid type
Word 2—Number of ACIDPs (integer)
Word 3—start of ACIDP list

ACIDP list format
Words 1..8—ACIDP name (16 ASCII characters)
Words 9..14—associated ACP name (12 ASCII characters)
Word 15—program status (integer)
0=NOT INST
1=TEST
2=RESTRICT
3=NORMAL
Word 16—execution state (integer)
0=ABORT
1=ACCESS
2=DELAY
3=OFF
4=RUN
5=WAIT
6=FAIL
Words 17..21—precollect DDT name (9 char, lsb unused)
Word 22—SCH trigger (integer) 0=OFF 1=ON
Word 23—PPS trigger (integer) 0=OFF 1=ON
Word 24—DMD trigger (integer) 0=OFF 1=ON

Use information:
There can be a maximum of 250 ACIDPs in a CG.

The information contained in list format words 17-24 is not available for display at a Universal Station.

4.3.8 CG Node List Return

This message is returned by the CG if the host processor has requested the CG node list. The information returned includes the configured values of a number of CG parameters.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 48
Word 2—Number of Words = 85
Word 5—CG Identifier = 3

Message format:
Word 1—Return status (integer) 0=normal status
1=invalid type
Word 2—LCN release number (integer)
Word 3—LCN revision number (future)
Word 4—CG node number (integer) 1..95
Word 5—CG version number (integer)
Word 6—CG revision number (integer)
Word 7—Time synch period (integer)
Word 8—Confirmation timeout period (integer)
Word 9—CG station address (integer) 1 or 3
Word 10—T1 Time Unit (integer)
Word 11—N2 Count (integer)
Word 12—Floating point format (ordinal)
Word 13—Baud rate (ordinal)
Word 14—Tagname length (boolean) 0=short (8 bytes)
1=long(16 bytes)
Word 15—History: user average period in minutes (integer).
Word 16—History: number of shifts in week (integer)
Word 17—History: week starting hour
Word 18—Month type (boolean) 0=calendar
1=28 days
Words 19..24—Reserved for future use
Word 25—LCN PIN node ID (2 ASCII characters)
Words 26..45—User comment (40 ASCII characters)
Words 46..85—Reserved for future use

Use information:
For additional information about the CG parameter values returned with this message, refer to the *CG Parameter Reference Dictionary*.

Sixteen-Character Tagname Form

Except for the Transaction Code (148), the sixteen-character tagname form for this message is identical to the eight-character tagname form.

4.4 PROGRAM SCHEDULING AND STATUS MESSAGES

4.4.1 ACIDP Program Mode Change Request

This message originated by the host processor, requests connection of an ACP to an ACIDP, and/or to change the PROGSTAT value. The allowed modes are: NOT INST, TEST, RESTRICT, and NORMAL.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 22
Word 2—Number of Words = 11
Word 5—CG Identifier = 3

Message format: Words 1..4—ACIDP name (8 ASCII characters)
Words 5..10—ACPROG (12 ASCII characters)
Word 11—PROGSTAT (integer) 0=not installed
1=test
2=restricted
3=normal

Use information: The assigned program status affects how the CG responds to requests from the associated ACP. Store data operations are permitted only if the program status is "normal"; other operations are permitted regardless of program status. See 2.3 for additional details.

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 122
Word 2—Number of Words = 20
Word 5—CG Identifier = 3

Message format: Words 1..8—ACIDP name (16 ASCII characters)
Words 9..13—reserved for future use (fill with integer zeros)
Words 14..19—ACPROG (12 ASCII characters)
Word 20—PROGSTAT (integer) 0=not installed
1=test
2=restricted
3=normal

Use information: The assigned program status affects how the CG responds to requests from the associated ACP. Store data operations are permitted only if the program status is "normal"; other operations are permitted regardless of program status. See 2.3 for additional details.

4.4.2 ACIDP Mode Change Return

This message originated by the CG signals success or failure of a requested program mode change.

Header information: Word 1—Transaction Code = 23
Word 2—Number of Words = 1
Word 5—CG Identifier = 3

Message format: Word 1—return status (integer) 0=normal return
1=ACP ACIDP mismatch
4=EXECSTAT incorrect
6=valid ACIDP not found

Use information: PROGSTAT changes can only be made while the ACIDP's EXECSTAT value is set to OFF, DELAY, or ABORT.

4.4.3 Get ACIDP Status Request

This message from the host processor is used to obtain all the parameter values for an ACIDP for display or other use at the host processor.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 24
Word 2—Number of Words = 4
Word 5—CG Identifier = 3

Message format: Words 1..4—ACIDP name (8 ASCII characters)

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 124
Word 2—Number of Words = 13
Word 5—CG Identifier = 3

Message format: Words 1..8—ACIDP name (16 ASCII characters)
Words 9..13—reserved for future use (fill with integer zeros)

4.4.4 Get ACIDP Status Return

This message from the CG returns all the parameter values for an ACIDP.

Eight-Character Tagname Form

Header information:

Word 1—Transaction Code = 25
Word 2—Number of Words = 1 (plus 67 if ACIDP found)
Word 5—CG Identifier = 3

Message format:

Word 1—return status (integer) 0=normal return
6=valid ACIDP not found

Words 2..5—ACIDP name (8 ASCII characters)
Words 6..17—PTDESC (24 ASCII characters)
Word 18—UNIT (2 ASCII characters)
Words 19..22—KEYWORD (8 ASCII characters)
Words 23..27—bytes 1-9: precollect DDT name (9 ASCII char),
byte 10: packed array of booleans, where

bit 7 (msb)	SCH trigger	1=ON	0=OFF
bit 6	PPS trigger	1=ON	0=OFF
bit 5	DMD trigger	1=ON	0=OFF

bits 4-0 are unused

Word 28—ACCESKEY (integer) 0=READONLY
1=READWRIT
Words 29..34—ACPROG (12 ASCII characters)
Word 35—GROUP ID (2 ASCII characters)
Word 36—ACT_TYPE (integer) 0=CYCLIC
1=PERIODIC
2=CYC_DMD
3=PER_DMD
4=DEMAND
Word 37—INH_STAT (integer) 0=INHIBIT
1=PERMIT
Word 38—EXECSTAT (integer) 0=ABORT
1=ACCESS
2=DELAY
3=OFF
4=RUN
5=WAIT
6=FAIL
Word 39—PROGSTAT (integer) 0=NOT INST
1=TEST
2=RESTRICT
3=NORMAL
Words 40..41—ABORTCOD (4 ASCII characters)
Word 42—not used
Word 43—not used
Word 44—CONFWAIT (integer)
Word 45—not used
Words 46..54—NEXT_RTM (17 ASCII characters)

Words 55..58—RTPERIOD (8 ASCII characters)
Words 59..62—STIME (8 ASCII characters)
Word 63—RUN_INIT (integer) 0=OFF, 1=ON
Word 64—OPER_DMD (integer) 0=OFF, 1=ON
Word 65—PPS (integer) 0=OFF, 1=ON
Word 66—not used
Word 67—CONF_RQD (integer) 0=OFF, 1=ON
Word 68—TAKE_IP (integer) 0=OFF, 1=ON

Use information: See the *Computer Gateway Parameter Reference Dictionary* for descriptions of these parameters, including the mechanisms for their setup and change. Most of these parameters can be viewed through standard displays at a Universal Station; see heading 6.3 for details.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 125
Word 2—Number of Words = 1 (plus 77 if ACIDP found)
Word 5—CG Identifier = 3

Message format:
Word 1—return status (integer) 0=normal return
6=valid ACIDP not found
Words 2..9—ACIDP name (16 ASCII characters)
Words 10..21—PTDESC (24 ASCII characters)
Word 22—UNIT (2 ASCII characters)
Words 23..26—KEYWORD (8 ASCII characters)
Words 27..31—bytes 1-9: precollect DDT name (9 ASCII char)
byte 10: packed array of booleans, where
bit 7 (msb) SCH trigger 1=ON 0=OFF
bit 6 PPS trigger 1=ON 0=OFF
bit 5 DMD trigger 1=ON 0=OFF
bits 4-0 are unused
Word 32—ACCESKEY (integer) 0=READONLY
1=READWRIT
Words 33..38—ACPROG (12 ASCII characters)
Word 39—GROUP ID (2 ASCII characters)
Word 40—RCASENB (0=OFF, 1=ON)
Words 41..45—reserved for future use
Word 46—ACT_TYPE (integer) 0=CYCLIC
1=PERIODIC
2=CYC_DMD
3=PER_DMD
4=DEMAND
Word 47—INH_STAT (integer) 0=INHIBIT
1=PERMIT
Word 48—EXECSTAT (integer) 0=ABORT
1=ACCESS
2=DELAY
3=OFF
4=RUN
5=WAIT
6=FAIL

Word 49—PROGSTAT (integer) 0=NOT INST
1=TEST
2=RESTRICT
3=NORMAL
Words 50..51—ABORTCOD (4 ASCII characters)
Word 52—MSG_INDX
Word 53—CONF_IND
Word 54—CONFWAIT (integer)
Word 55—not used
Words 56..64—NEXT_RTM (18 ASCII characters)
Words 65..68—RTPERIOD (8 ASCI characters)
Words 69..72—STIME (8 ASCII characters)
Word 73—RUN_INIT (integer) 0=OFF, 1=ON
Word 74—OPER_DMD (integer) 0=OFF, 1=ON
Word 75—PPS (integer) 0=OFF, 1=ON
Word 76—MSG_CONF
Word 77—CONF_RQD (integer) 0=OFF, 1=ON
Word 78—TAKE_IP (integer) 0=OFF, 1=ON

Use information:

See the *Computer Gateway Parameter Reference Dictionary* for descriptions of these parameters, including the mechanisms for their setup and change. Most of these parameters can be viewed through standard displays at a Universal Station; see heading 6.3 for details.

4.4.5 Turn On ACP

This is the only unsolicited application-level message from the CG during normal operation and is used to signal the need for a specified ACP to begin execution.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 2
Word 2—Number of Words = 9
Word 5—CG Identifier = 1
Word 6—Host Task Identifier = DI

Message format:
Words 1..4—ACIDP name (8 ASCII characters)

Word 5—OPER_DMD (integer)	0=OFF, 1=ON
Word 6—PPS (integer)	0=OFF, 1=ON
Word 7—TAKE_I_P (integer)	0=OFF, 1=ON
Word 8—SCHEDULED (integer)	0=OFF, 1=ON
Word 9—PS_MSG (integer)	0=OFF, 1=ON

Use information:
The CG resets each of these values to OFF upon dispatch of this message in order to prepare for the next "trigger" event (which may occur while the ACP is running). The host processor must provide for the event of this message arriving while the associated ACP is executing.

The ACP needs to examine this data to determine why it was activated. SCHEDULED is set ON whenever the ACP is to be executed because of either CYCLIC or PERIODIC scheduling. PS_MSG is set ON whenever a message for this ACP is being held at the CG. PPS is defined at heading 2.2.1.3. OPER_DMD, and TAKE_I_P are defined at paragraph 6.4.

The host processor reaction to this message should include

- Activation of the addressed ACP (passing to it the message information contained in words 5 through 9.
- Dispatch of an ACIDP execution state change (to run) to the CG.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 102
Word 2—Number of Words = 18
Word 5—CG Identifier = 1
Word 6—Host Task Identifier = DI

Message format:
Words 1..8—ACIDP name (16 ASCII characters)
Words 9..13—reserved for future use (fill with integer zeros)
Word 14—OPER_DMD (integer) 0=OFF, 1=ON
Word 15—PPS (integer) 0=OFF, 1=ON
Word 16—TAKE_I_P (integer) 0=OFF, 1=ON
Word 17—SCHEDULED (integer) 0=OFF, 1=ON
Word 18—PS_MSG (integer) 0=OFF, 1=ON

Use information:
The CG resets each of the parameter values represented by words 14 through 18 to OFF upon dispatch of this message in order to prepare for the next "trigger" event (which may occur while the ACP is running). The host processor must provide for the event of this message arriving while the associated ACP is executing.

The ACP needs to examine this data to determine why it was activated. SCHEDULED is set ON whenever the ACP is to be executed because of either CYCLIC or PERIODIC scheduling. PS_MSG is set ON whenever a message for this ACP is being held at the CG. PPS is defined at heading 2.2.1.3. OPER_DMD, and TAKE_I_P are defined at paragraph 6.4.

The host processor reaction to this message should include

- Activation of the addressed ACP (passing to it the message information contained in words 5 through 9.
- Dispatch of an ACIDP execution state change (to run) to the CG.

Note that words 19 through 76 are variable in length and are significant to certain revisions of CM50S and CM50N. Thus, the size of the message (contained in header word 2) can vary from 18 to 76 words.

4.4.6 ACIDP Execution State Change Request

This message, originated by the host processor, informs the CG of a change of an ACP's execution state. It does this by requesting a change of the ACP execution state as recorded in the parameter EXECSTAT of its associated ACIDP. The allowable operations are to change to RUN state or from RUN to ABORT or to OFF/DELAY. The CG sets the ACIDP to OFF if its activation type is DEMAND, and to DELAY if the activation type is PERIODIC, CYCLIC, PERIODIC/DEMAND, or CYCLIC/DEMAND.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 21
Word 2—Number of Words = 7
Word 5—CG Identifier = 3

Message format:
Words 1..4—ACIDP name (8 ASCII characters)
Word 5—new state (integer) 1=off/delay state
2=abort state
3=run state
Words 6..7—abort code (4 ASCII characters)

Use information:
The abort codes are visible at the Universal Station and can have any value assignments you wish. Presumably they will be coded to indicate the apparent reason for aborting the ACP/ACIDP. Abort code of 0000 (ASCII zeros) indicates a normal termination.

The change to RUN state message is required after receipt by the host processor of a turn-on message. This confirms to the CG that the ACP has been activated. If the ACP is connected to an ACIDP, the change to RUN state is required prior to use of any messages addressed to the ACIDP.

There is no data-message response to this request.

The operator can change the EXESTAT of an ACIDP from ABORT to OFF/DELAY through the operator-demand function at the Universal Station.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 121
Word 2—Number of Words = 16
Word 5—CG Identifier = 3

Message format:
Words 1..8—ACIDP name (16 ASCII characters)
Words 9..13—reserved for future use (fill with integer zeros)
Word 14—new state (integer) 1=off/delay state
2=abort state
3=run state
Words 15..16—abort code (4 ASCII characters)

Use information:

The abort codes are visible at the Universal Station and can have any value assignments you wish. Presumably they will be coded to indicate the apparent reason for aborting the ACP/ACIDP. Abort code of 0000 (ASCII zeros) indicates a normal termination.

The change to RUN state message is required after receipt by the host processor of a turn-on message. This confirms to the CG that the ACP has been activated. If the ACP is connected to an ACIDP, the change to RUN state is required prior to use of any messages addressed to the ACIDP.

There is no data-message response to this request.

The operator can change the EXESTAT of an ACIDP from ABORT to OFF/DELAY through the operator-demand function at the Universal Station.

4.4.7 Internal DDT Connect or Disconnect Request

The host processor sends this request to initiate the connect or disconnect of a CG-resident Internal DDT (input values only) to a specified ACIDP that already has a connected ACP. It also is used to change "trigger" values for a previously connected DDT.

This DDT to ACIDP connection enables automatic precollection of data from points specified in the DDT whenever the ACP is to be initiated as the result of a "trigger" event (schedule, point process special, or operator demand). See paragraph 3 in Appendix C.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 45
Word 2—Number of Words = 19
Word 5—CG Identifier = 3

Message format:
Word 1—action requested (integer) 0=connect
1=disconnect
2=change triggers
Word 2—Schedule trigger (integer) 0=OFF, 1=ON
Word 3—Point Process Special trigger (integer) 0=OFF, 1=ON
Word 4—Operator Demand trigger (integer) 0=OFF, 1=ON
Words 5..8—ACIDP name (8 ASCII characters)
Words 9..14—ACPROG (12 ASCII characters)
Words 15..19—DDT name (9 ASCII characters + unused byte)

Use information:
Same as for the sixteen-character tagname form explained below.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 145
Word 2—Number of Words = 28
Word 5—CG Identifier = 3

Message format:
Word 1—action requested (integer) 0=connect
1=disconnect
2=change triggers
Word 2—Schedule trigger (integer) 0=OFF, 1=ON
Word 3—Point Process Special trigger (integer) 0=OFF, 1=ON
Word 4—Operator Demand trigger (integer) 0=OFF, 1=ON
Words 5..9—reserved for future use (fill with integer zeros)
Words 10..17—ACIDP name (16 ASCII characters)
Words 18..23—ACPROG (12 ASCII characters)
Words 24..28—DDT name (9 ASCII characters + unused byte)

Use information: Precollection occurs only for the ON trigger or triggers.

In order for the request to be accepted, there must be a correlation between the message data and what already exists in the CG database. The following table summarizes these relationships as they affect the request type.

	CONNECT	DISCONNECT	CHANGE TRIGGERS
At least one trigger type set ON	must	don't care	must
ACIDP-ACP connection match	must	must	must
DDT is CG-resident	must	must	must
No existing DDT-ACIDP connection	must	invalid	invalid
DDT-ACIDP connection matches	error	must	can
DDT-ACIDP connection matches, but request has different trigger values	error	triggers don't matter	can

4.4.8 Internal DDT Connect or Disconnect Return

This message originated by the CG signals success or failure of a requested DDT connect, disconnect, or trigger values change.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 46
Word 2—Number of Words = 20
Word 5—CG Identifier = 3

Message format:
Word 1—return status (integer) 0=normal return
1=nonexistent table
6=valid ACIDP not found
18=DDT already connected
19=DDT-ACIDP not connected
20=ACP-ACIDP not connected
21=invalid option value
Word 2—action performed (integer) 0=connect
1=disconnect
2=change trigger
Word 3—SCH trigger value (integer) 0=OFF, 1=ON
Word 4—PPS trigger value (integer) 0=OFF, 1=ON
Word 5—DMD trigger value (integer) 0=OFF, 1=ON
Words 6..9—ACIDP name (8 ASCII characters)
Words 10..15—ACPROG (12 ASCII characters)
Words 16..20—DDT name (9 ASCII characters, + unused byte)

Use information:
The values returned are those to be found in the CG database after the command is serviced. Even if the request is invalid, the CG returns whatever data exists for that ACIDP.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 146
Word 2—Number of Words = 29
Word 5—CG Identifier = 3

Message format:
Word 1—return status (integer) 0=normal return
1=nonexistent table
6=valid ACIDP not found
18=DDT already connected
19=DDT-ACIDP not connected
20=ACP-ACIDP not connected
21=invalid option value
Word 2—action performed (integer) 0=connect
1=disconnect
2=change trigger
Word 3—SCH trigger value (integer) 0=OFF, 1=ON
Word 4—PPS trigger value (integer) 0=OFF, 1=ON
Word 5—DMD trigger value (integer) 0=OFF, 1=ON
Words 6..10—reserved for future use (fill with integer zeros)
Words 11..18—ACIDP name (16 ASCII characters)
Words 19..24—ACPROG (12 ASCII characters)
Words 25..29—DDT name (9 ASCII characters, + unused byte)

Use information:
The values returned are those to be found in the CG database after the command is serviced. Even if the request is invalid, the CG returns whatever data exists for that ACIDP.

4.5 INTERNAL DDT PREPARATION MESSAGES

The messages in this section are a necessary precursor of any table-driven get and store requests. They are concerned with the establishment and saving of point-identifier tables required by the table-driven request messages.

4.5.1 Build Internal DDT Request

The host processor sends this request to initiate the external-to-internal identifier conversions for a specified table of point.parameters.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 6
Word 2—Number of Words = 9 + total for external name/parameter sets (10 words each)
Word 5—CG Identifier = 2

Message format:
Word 1—table purpose (integer) 1=input table
2=output table
5=history query table

Words 2..5—order of data types in table (integer)
0=no entries
1=real
2=integer
3=ASCII (24-character string)
4=enumeration* (ASCII string) OR
5=enumeration (ordinal value)
13=internal entity id
15=external entity id (ASCII name)**
17=LCN internal time
19=ASCII (40-character string)

These ordering values can be assigned in any sequence; however, none of the data types can be repeated, and you cannot have both enumeration and ordinal types in the same DDT.

Words 6..9—number of points of each type (integer array)

These values must be presented in the same order as in words 2 through 5.

Name/param set format:
Word 10—start of 10-word external name/parameter sets
Words 1..4—point name (8 ASCII characters)
Word 5—point name index (integer)
Words 6..9—parameter name (ASCII characters)
Word 10—parameter index (integer)

Use information:
Same as for the sixteen-character tagname form explained below.

*See Paragraph 4.6.4 for restrictions on the use of Enumeration values in Store Data requests.
**Store (output) of external entity id (ASCII name) is not permitted.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 106
Word 2—Number of Words = 9 + total for external name/parameter sets (21 words each)
Word 5—CG Identifier = 2

Message format:
Word 1—start of external data table

Data table format:
Word 1—table purpose (integer)
1=input table
2=output table
5=history query table

Words 2..5—order of data types in table (integer)
0=no entries
1=real
2=integer
3=ASCII (24-character string)
4=enumeration* (ASCII string) OR
5=enumeration (ordinal value)
13=internal entity id
15=external entity id (ASCII name)**
17=LCN internal time
19=ASCII (40-character string)

These ordering values can be assigned in any sequence; however, none of the data types can be repeated, and you cannot have both enumeration and ordinal types in the same DDT.

Words 6..9—number of points of each type (integer array)

These values must be presented in the same order as in words 2 through 5.

Word 10—start of 10-word external name/parameter sets

Name/param set format:
Word 1—LCN PIN node ID (2 ASCII characters)
Words 2..9—point name (16 ASCII characters)
Word 10—point name index (integer)
Words 11..14—parameter name (8 ASCII characters)
Word 15—parameter index x (integer)
Word 16—reserved for future parameter index y (integer)
Words 17..21—reserved for future use (fill with integer zeros)

*See Paragraph 4.6.4 for restrictions on the use of Enumeration values in Store Data requests.
**Store (output) of external entity id (ASCII name) is not permitted.

Use information:

"Point name index" must = 0

"Parameter name index" is used to identify a specific parameter array element. When used with nonarray parameters, this field must = 0.

The order and number of points for each data type is significant (to the CG for processing of Store Data messages and to the ACP for the processing of Get Data Return messages) because the number of data words for each point varies with data type.

For history requests, "order of data types" = (0, 1, 2, 5) and "number of points of each type" = total number of history points identified.

4.5.2 Build Internal DDT Return

The CG sends this response, which includes data corresponding to the build Internal DDT request.

Header information:
Word 1—Transaction Code = 9
Word 2—Number of Words = size of the Internal DDT plus one
Word 5—CG Identifier = 2

Message format:
Word 1—return status (integer) 0=normal status
5=complete with errors
7=data access failure
Word 2—start of Internal DDT

Data Table Formats:
Word 1—total number of words in this table (integer)

Word 2—table purpose (integer) 1=input table
2=output table
5=history query table
Words 3..6—order of data types in table (integer)
0=no entries
1=real
2=integer
3=ASCII (24-character string)
4=enumeration (ASCII string) OR
5=enumeration (ordinal value)
13=internal entity id
15=external entity id (ASCII name)
17=LCN internal time
19=ASCII (40-character string)
Words 7..10—number of points of each type (integer array)
Word 11—start of 6-word internal name/parameter sets

Name/parameter set format–normal (Word 1 not = 255)

Words 1..4—internal point identifier (4 integer values)
Word 5—internal parameter identifier (integer)
Word 6—parameter subscript (integer)

For Output tables only: The sets of 6-word internal point and parameter identifiers are followed by a number of additional 16-bit integer words equal to the number of type 4 or 5 elements. These words hold the internal set identifier for each enumeration and must be part of the IDB stored for future use.

Name/parameter set format–point error

Word 1—255 (integer)
Words 2..3—not used
Word 4—error code (integer)
1=Point name error (entity name error)
2=Duplicate point name (duplicate entity error)
3=Incomplete search
4=Subscript out of range
5=Point does not have subscript
6=Parameter name error
7=Invalid parameter for point name type
8=Enumeration fetch error
9=Point is not Boolean or Enumeration
Words 5..6—not used

Use information: If the return status is other than 0, the Internal DDT is not usable, and the build request must be tried again. If the return status is 5, the external table may need modification. See the returned error information codes.

4.5.3 Store Internal DDT in CG Request

This message from the host processor sends a copy of a Internal DDT for the CG to use in Data Table-driven Get Data requests.

Header information: Word 1—Transaction Code = 10
Word 2—Number of Words = 5 plus size of data table
Word 5—CG Identifier = 3

Message format: Words 1..5—table identifier (9 ASCII characters)
Word 6—start of data table, see paragraph 4.5.2 for format

Use information: CG-resident Internal DDTs cannot be used by Store Data or Get History.

4.5.4 Delete Internal DDT Table from CG Request

This message from the host processor requests that a previously installed Internal DDT table be deleted from the CG.

Header information: Word 1—Transaction Code = 11
Word 2—Number of Words = 5
Word 5—CG Identifier = 3

Message format: Words 1..5—table identifier (9 ASCII characters)

4.5.5 Internal DDT Store/Delete Return

This message from the CG is in response to either a Internal DDT Store or a Internal DDT Delete request.

Header information: Word 1—Transaction Code = 12
Word 2—Number of Words = 1
Word 5—CG Identifier = 3

Message format: Word 1—return status (integer)
0=successful completion
1=nonexistent table
2=table space not available
4=table name used
5=memory not available
8=table in use
33=DDT—bad data type
34=DDT—bad number of words
35=DDT—bad table purpose
36=DDT—bad table size
37=DDT—duplicate data type
38=DDT—data type mismatch
39=DDT—too many entries
40=header—bad number of words

4.6 POINT-DATA TRANSFERS USING DATA TABLES

The following group of data-transfer messages require the use of previously built and converted Internal Data Tables (Internal DDT). See paragraph 4.5 for descriptions of the messages used in Internal DDT preparation, and for table format and content.

4.6.1 Get Data Request (Internal DDT in CG)

This message from the host processor is used when a copy of the Internal DDT is kept by the CG (see paragraph 4.5.3).

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 3
Word 2—Number of Words = 10
Word 5—CG Identifier = 2

Message format: Words 1..4—ACIDP name or blank (8 ASCII characters)

Word 5—data access priority (integer) 1=control
2=noncontrol

Words 6..10—Internal DDT name (9 ASCII characters followed by a blank character)

Use information: The referenced Internal DDT must be previously created (messages 6 and 9) and stored in the CG database (messages 10 and 12) with no errors reported.

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 103
Word 2—Number of Words = 19
Word 5—CG Identifier = 2

Message format: Words 1..8—ACIDP name or blank (16 ASCII characters)
Words 9..13—reserved for future use
Word 14—data access priority (integer) 1=control
2=noncontrol
Words 15..19—Internal DDT name (9 ASCII characters followed by a blank character)

Use information: The referenced Internal DDT must be previously created (messages 106 and 9) and stored in the CG database (messages 10 and 12) with no errors reported.

4.6.2 Get Data Request (Internal DDT in Host Processor)

This message from the host processor is used when no copy of the Internal DDT is kept by the CG, and the Internal DDT becomes part of the request message.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 4
Word 2—Number of Words = 5 plus size of the Internal DDT
Word 5—CG Identifier = 2

Message format: Words 1..4—ACIDP or blank (8 ASCII characters)

Word 5—data access priority (integer) 1=control
2=noncontrol

Word 6—start of Internal DDT formatted as explained in paragraph 4.5.2

Use information: The Internal DDT used by this request must be previously created (messages 6 and 9) with no errors reported.

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 104
Word 2—Number of Words = 14 plus size of the Internal DDT
Word 5—CG Identifier = 2

Message format: Words 1..8—ACIDP or blank (16 ASCII characters)
Words 9..13—reserved for future use (fill with integer zeros)
Word 14—data access priority (integer) 1=control
2=noncontrol
Word 15—start of Internal DDT formatted as explained in paragraph 4.5.2

Use information: The Internal DDT used by this request must be previously created (messages 106 and 9) with no errors reported.

4.6.3 Get Data Return

This message from the CG is in response to either type of Get Data request (message types 3 and 4). The data returned includes both values and value status. The number of words returned for each point varies by data type (3 for Reals, 2 for Integers, 13 for ASCII strings, and 5 for Enumerations).

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 7
Word 2—Number of Words = 5 plus size of data
Word 5—CG Identifier = 2

Message format:
Words 1..4—ACIDP name or blank (8 ASCII characters)
Word 5—return status (integer)

0=normal
1=nonexistent table
4=EXECSTAT incorrect
5=complete with errors
10=get-store table empty
16=ACIDP name invalid
28=invalid data access priority
33=invalid data type specified in Internal DDT
35=Internal DDT purpose not same as request
36=Internal DDT header number of words disagrees with table content
37=data type repeated in Internal DDT
38=Internal DDT contains both enumeration and ordinal data
39=Internal DDT contains over 300 points
40=number of words specified in message header disagrees with message content

Word 6—start of values for point/parameters (in same order as contained in the Internal DDT)

Parameter value formats:

Real	Words 1 and 2—value (real) Word 3—value status (integer)
Integer	Word 1—value (integer) Word 2—value status (integer)
ASCII String (24 characters)	Words 1..12—value (ASCII characters) Word 13—value status (integer)
Enumeration	Words 1..4—value (ASCII characters) Word 5—value status (integer)
Ordinal Value of Enumeration	Word 1—value (integer) Word 2—value status (integer)

Internal Entity ID	Words 1..4—value (internal format) Word 5—value status (integer)
External Entity ID	Words 1..8—name (up to 16 ASCII char.) Word 9—LCN id (2 characters; blank=local) Word 10—value status (integer)
LCN Internal Time	Words 1..2—seconds since 1 January 1979 (32-bit integer) Word 3—tenths of ms in current second (integer) Word 4—value status (integer)
ASCII String (40 characters)	Words 1..20—value (ASCII characters) Word 21—value status (integer)

Use information: The number of point/parameters is limited to a total of 300 or less. See heading 4.1.5.

For value status definitions, see "Data Access Status Codes" in the *Messages Directory*.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 107
Word 2—Number of Words = 14 plus size of data
Word 5—CG Identifier = 2

Message format:
Words 1..8—ACIDP name or blank (16 ASCII characters)
Words 9..13—reserved for future use (fill with integer zeros)
Word 14—return status (integer)

- 0=normal
- 1=nonexistent table
- 4=EXECSTAT incorrect
- 5=complete with errors
- 10=get-store table empty
- 16=ACIDP name invalid
- 28=invalid data access priority
- 33=invalid data type specified in Internal DDT
- 35=Internal DDT purpose not same as request
- 36=Internal DDT header number of words disagrees with table content
- 37=data type repeated in Internal DDT
- 38=Internal DDT contains both enumeration and ordinal data
- 39=Internal DDT contains over 300 points
- 40=number of words specified in message header disagrees with message content

Word 6—start of values for point/parameters (in same order as contained in the Internal DDT)

Parameter value formats:

Real	Words 1 and 2—value (real) Word 3—value status (integer)
Integer	Word 1—value (integer) Word 2—value status (integer)
ASCII String (24 characters)	Words 1..12—value (ASCII characters) Word 13—value status (integer)
Enumeration	Words 1..4—value (ASCII characters) Word 5—value status (integer)
Ordinal Value of Enumeration	Word 1—value (integer) Word 2—value status (integer)
Internal Entity ID	Words 1..4—value (internal format) Word 5—value status (integer)
External Entity ID	Words 1..8—name (up to 16 ASCII char.) Word 9—LCN id (2 characters; blank=local) Word 10—value status (integer)
LCN Internal Time	Words 1..2—seconds since 1 January 1979 (32-bit integer) Word 3—tenths of ms in current second (integer) Word 4—value status (integer)
ASCII String (40 characters)	Words 1..20—value (ASCII characters) Word 21—value status (integer)

Use information:

The number of point/parameters is limited to a total of 300 or less. See heading 4.1.5.

For value status definitions, see “Data Access Status Codes” in the *Messages Directory*.

4.6.4 Store Data Request

This message from the host processor is used to store values to locations in the CG and other LCN devices.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 5
Word 2—Number of Words = 5 plus size of data
Word 5—CG Identifier = 2

Message format:
Words 1..4—ACIDP name (8 ASCII characters)
Word 5—data access priority (integer) l=control
2=noncontrol
Word 6—start of Internal DDT table (see paragraph 4.5)
Word 6+Internal DDT table length—start of parameter values (in same order as in Internal DDT).

Parameter value formats:

Real	Words 1 and 2—value (real) Word 3—store status (integer)
Integer	Word 1—value (integer) Word 2—store status (integer)
ASCII	Words 1..12—value (ASCII characters) Word 13—store status (integer)
Enumeration	Words 1..4—value (enumeration) Word 5—store status (integer)
Ordinal Value of Enumeration	Word 1—value (integer) Word 2—store status (integer)
Internal Entity ID	Words 1..4—value (internal format) Word 5—value status (integer)
LCN Internal Time	Words 1..2—seconds since 1 January 1979 (32-bit integer) Word 3—tenths of ms in current second (integer) Word 4—value status (integer)
ASCII String (40 characters)	Words 1..20—value (ASCII characters) Word 21—value status (integer)

Use information: The number of point/parameters is limited to a total of 300 or less. See heading 4.1.5. The Internal DDT used by this request must be previously created (messages 6 and 9) with no errors reported.

Because of the possibility for errors in point and parameter definition, caution in use of this message type is advised. One suggested technique is to read and check the data values before attempting to change them.

The store status controls what—if any—value is to be stored.
0=Store the data value provided
1=Store the bad value representation instead
2=Do not store any value
Store status value of 1 is valid only for Real or ASCII data. The bad value representations are NaN for Real values and question mark strings for ASCII.

The result from a store status of 1 depends on the point type. For reals and integers, NaN (-0) is stored. For ASCII values and enumerations, strings of question marks are stored.

The "do not store" (store status = 2) allows for selective storing to a portion of the points named by the Internal DDT. Return status for these points is set to 255 (Directed no store).

For further information on standard enumerations, see the
Hiway Gateway Parameter Reference Dictionary.
Process Manager Parameter Reference Dictionary.
Computer Gateway Parameter Reference Dictionary.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 105
Word 2—Number of Words = 14 plus size of data
Word 5—CG Identifier = 2

Message format:
Words 1..8—ACIDP name (16 ASCII characters)
Words 9..13—reserved for future use (fill with integer zeros)
Word 14—data access priority (integer) l=control
2=noncontrol
Word 15—start of Internal DDT table (see paragraph 4.5)
Word 15+Internal DDT table length—start of parameter values (in same order as in Internal DDT).

Parameter value formats:

Real	Words 1 and 2—value (real) Word 3—store status (integer)
Integer	Word 1—value (integer) Word 2—store status (integer)

ASCII	Words 1..12—value (ASCII characters) Word 13—store status (integer)
Enumeration	Words 1..4—value (enumeration) Word 5—store status (integer)
Ordinal Value of Enumeration	Word 1—value (integer) Word 2—store status (integer)
Internal Entity ID	Words 1..4—value (internal format) Word 5—value status (integer)
LCN Internal Time	Words 1..2—seconds since 1 January 1979 (32-bit integer) Word 3—tenths of ms in current second (integer) Word 4—value status (integer)
ASCII String (40 characters)	Words 1..20—value (ASCII characters) Word 21—value status (integer)

Use information:

The number of point/parameters is limited to a total of 300 or less. See heading 4.1.5. The Internal DDT used by this request must be previously created (messages 6 and 9) with no errors reported.

Because of the possibility for errors in point and parameter definition, caution in use of this message type is advised. One suggested technique is to read and check the data values before attempting to change them.

The store status controls what—if any—value is to be stored.
0=Store the data value provided
1=Store the bad value representation instead
2=Do not store any value
Store status value of 1 is valid only for Real or ASCII data. The bad value representations are NaN for Real values and question mark strings for ASCII.

The result from a store status of 1 depends on the point type. For reals and integers, NaN (-0) is stored. For ASCII values and enumerations, strings of question marks are stored.

The "do not store" (store status = 2) allows for selective storing to a portion of the points named by the Internal DDT. Return status for these points is set to 255 (Directed no store).

This message type can store character-string values to standard enumerations, but not to self-defined enumerations. Storing of ordinal values to self-defined enumerations is permitted. For further information on standard enumerations, see the *Hiway Gateway Parameter Reference Dictionary*, the *Process Manager Parameter Reference Dictionary*, and the *Computer Gateway Parameter Reference Dictionary.*

4.6.5 Store Data Return

This message from the CG is in response to a store data request.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 8
Word 2—Number of Words = 5 plus size of tables
Word 5—CG Identifier = 2

Message format: Words 1..4—ACIDP name (8 ASCII characters)
Word 5—return status (integer)

0=normal
4=EXECSTAT incorrect
5=complete with errors
6=valid ACIDP not found
8=access key incorrect
15=incorrect EXECSTAT
25=data must be enumeration type
28=invalid data access priority
33=invalid data type specified in Internal DDT
35=Internal DDT purpose not same as request
36=Internal DDT header number of words disagrees with table content
37=data type repeated in Internal DDT
38=Internal DDT contains both enumeration and ordinal data
39=Internal DDT contains over 300 points
40=number of words specified in message header disagrees with message content

Word 6—start of return status per point (integer array)

Use information: For value status definitions, see “Data Access Status Codes” in the *Messages Directory*.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 108
Word 2—Number of Words = 14 plus size of tables
Word 5—CG Identifier = 2

Message format:
Words 1..8—ACIDP name (16 ASCII characters)
Words 9..13—reserved for future use
Word 14—return status (integer)
0=normal
4=EXECSTAT incorrect
5=complete with errors
6=valid ACIDP not found
8=access key incorrect
15=incorrect EXECSTAT
25=data must be enumeration type
28=invalid data access priority
33=invalid data type specified in Internal DDT
35=Internal DDT purpose not same as request
36=Internal DDT header number of words disagrees with table content
37=data type repeated in Internal DDT
38=Internal DDT contains both enumeration and ordinal data
39=Internal DDT contains over 300 points
40=number of words specified in message header disagrees with message content
Word 15—start of return status per point (integer array)

Use information:
For value status definitions, see “Data Access Status Codes” in the *Messages Directory*.

4.7 SINGLE-POINT DATA TRANSFERS

Single-point data transfers are used to get or store a value for a single parameter, for a selected parameter within a parameter array, or for all elements of a parameter array. (Parameter arrays are used with Custom Data Segments.)

Two methods are provided for single parameter gets and stores. The first allows you to use the external name that requires the CG to make a separate external to internal point-name conversion for each get/store call (requiring a search of other LCN devices to get this information). The second method allows a program in the host processor to make a name conversion call once (for each point) and then use the internal name with the (quicker) get/store call.

The internal IDs returned to the user program by the single point convert call are the actual LCN internal entity IDs. You need to take every possible precaution to protect this data from corruption and you should not attempt to pass altered entity IDs to the CG for any reason.

Single-point data transfers should not be used if large numbers of points are involved. In those cases, use the table-driven requests instead.

4.7.1 Processing of Parameters and Parameter Arrays

The processing of parameters and parameter arrays deserves some special attention because the single-point access to these values is not intuitively obvious.

The key is in two fields found in most single-point requests: "data type" and "parameter array pointer/size."

To get or store all elements of a parameter array, use the appropriate array “data type” with the value of “parameter array pointer/size” equal to array size.

To get only part of an array, use the appropriate array “data type” with the value of “parameter array pointer/size” equal to the number of elements desired. If the “parameter array pointer/size” is greater than the size of the array, the whole array will be returned.

Single-Point stores of “data type” 20 (ASCII array) may store part of an array, if the “parameter array pointer/size” (size of array being stored) is smaller than the actual array’s size.

Storing any other array “data type” with “parameter array pointer/size” not equal to the actual array’s size will result in an error.

4.7.1.1 Array Length Restrictions

The maximum number of elements that can be transferred for each array data type is:

• Real Array	1024	• Internal Entity Identifier Array	1000
• Integer Array	1024	• External Entity Identifier Array	500
• Enumeration Array	1024	• LCN Internal Time Array	1024
• Ordinal Values Array	1024	• ASCII Array	152

4.7.2 Get Single Point Request (External Identifier)

This request from the host processor is used to fetch a single value (or an array of values) for a point and parameter specified in the ACP's calling sequence, using the external identifier.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 28
Word 2—Number of Words = 16
Word 5—CG Identifier = 2

Message format:
Words 1..4—ACIDP name or blank (8 ASCII characters)
Word 5—data access priority (integer) 1=control
2=noncontrol
Word 6—data type (integer) 1=real
2=integer
3=ASCII (24 character string)
4=enumeration/boolean
5=enumeration ordinal value
6=LCN external time (see 4.7.3)
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
15=external entity id (ASCII name)
16=external entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40 character string)
20=ASCII (40 char. string) array
Words 7..10—point name (8 ASCII characters)
Word 11—point name index (integer)
Words 12..15—parameter name (8 ASCII characters)
Word 16—parameter array pointer/size (integer)

Use information:
Point name index must = 0

Content of the field "parameter array pointer/size" depends on data-type specification. See paragraph 4.7.1 for an explanation of this field's use with single parameters and with parameter arrays.

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 128
Word 2—Number of Words = 31
Word 5—CG Identifier = 2

Message format: Words 1..8—ACIDP name or blank (16 ASCII characters)
Word 9—data access priority (integer) 1=control
2=noncontrol
Word 10—data type (integer) 1=real
2=integer
3=ASCII (24 character string)
4=enumeration/boolean
5=enumeration ordinal value
6=LCN external time (see 4.7.3)
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
15=external entity id (ASCII name)
16=external entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40 character string)
20=ASCII (40 char. string) array
Word 11—LCN PIN node ID (2 ASCII characters)
Words 12..19—point name (16 ASCII characters)
Word 20—point name index (integer)
Words 21..24—parameter name (8 ASCII characters)
Word 25—parameter index x (integer)
Word 26—parameter index y (future use)
Words 27..31—reserved for future use (fill with integer zeros)

Use information: Point name index must = 0

Content of the field "parameter array pointer/size" depends on data-type specification. See paragraph 4.7.1 for an explanation of this field's use with single parameters and with parameter arrays.

4.7.3 Get LCN Time

This variation of the get single point request (28 or 128) is used to fetch the LCN time in a format of MM/DD/YY HH:MM:SS. The header is unchanged. The data type is specified as 6 (time type). The remaining fields are ignored.

4.7.4 Get Single Point Return (External Identifier)

This message from the CG is the response to a Get Single Point request (external identifier). The data returned includes both value and value status.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 29
Word 2—Number of Words = 6 plus size of data record
Word 5—CG Identifier = 2

Message format: Words 1..4—ACIDP name or blank (8 ASCII characters)
Word 5—return status (integer)

0=normal
5=check value status
6=entity not found
7=data access failure
(complete error logged in SMCC display)
11=data type mismatch
(data type corrected, no data returned)
15=incorrect execution state
16=ACIDP name invalid
20=ACP-ACIDP not connected
22=data type out of range
26=parameter not found
27=parameter index out of range
28=bad data access priority
29=bad data type
30=bad entity index
31=bad entity-parameter combination
32=bad parameter index
40=header bad number words
41=array size too large
(size of whole array and data returned)
42=Get LCN Time problems
44=duplicate entity name error

NOTE

Return Status codes 11 and 41 indicate warnings. Useful information has been returned. Return Status Code 5 indicates that the value status contains a non-normal data access status code.

Word 6—data type (integer)
1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
6=LCN external time (see 4.7.3)
7=real array
8=integer array
9=enumeration array
10=enum ordinal values array
13=internal entity id
14=internal entity id array
15=external entity id (ASCII name)
16=external entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array

Word 7—start of data record

Data record formats:

Real	Words 1 and 2—value (real) Word 3—value status (integer)
Integer	Word 1—value (integer) Word 2—value status (integer)
ASCII String (24 characters)	Words 1..12—value (ASCII characters) Word 13—value status (integer)
Enumeration	Words 1..4—value (ASCII characters) Word 5—value status (integer)
Ordinal	Word 1—ordinal value of enum (integer) Word 2—value status (integer)
Time	Words 1..9—time value (ASCII characters) Word 10—value status (integer)
Real Array	Word 1—array value status (integer) Words 2..3—array element 1 (real) Words 4..5—array element 2 (real) etc.
Integer Array	Word 1—array value status (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Enum Array	Word 1—array value status (integer) Words 2..5—array element 1 (ASCII char) Words 6..9—array element 2 (ASCII char) etc.

Ordinal Array	Word 1—array value status (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Internal Entity ID	Words 1..4—value (internal format) Word 5—value status (integer)
Internal Entity ID Array	Word 1—value status (integer) Words 2..5—array element 1 Words 6..9—array element 2 etc.
External Entity ID	Words 1..8—name (up to 16 ASCII char) Word 9—LCN id (2 characters; blank=local) Word 10—value status (integer)
External Entity ID Array	Word 1—value status (integer) Words 2..10—array element 1 Words 11..19—array element 2 etc.
LCN Internal Time	Words 1..2—seconds since 1 January 1979 (32-bit integer) Word 3—tenths of ms in current second (integer) Word 4—value status (integer)
LCN Internal Time Array	Word 1—value status (integer) Words 2..4—array element 1 Words 5..7—array element 2 etc.
ASCII String (40 characters)	Words 1..20—value (ASCII characters) Word 21—value status (integer)
ASCII String (40 characters) Array	Word 1—value status (integer) Words 2..21—array element 1 Words 22..41—array element 2 etc.

Use information: When data type is an array, the value status applies to the whole array. For value status definitions, see “Data Access Status Codes” in the *Messages Directory*.

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 129
Word 2—Number of Words = 15 plus size of data record
Word 5—CG Identifier = 2

Message format: Words 1..8—ACIDP name or blank (16 ASCII characters)
Word 9..13—reserved for future use (fill with integer zeros)
Word 14—return status (integer)

0=normal
5=check value status
6=entity not found
7=data access failure
(complete error logged in SMCC display)
11=data type mismatch
(data type corrected, no data returned)
15=incorrect execution state
16=ACIDP name invalid
20=ACP-ACIDP not connected
22=data type out of range
26=parameter not found
27=parameter index out of range
28=bad data access priority
29=bad data type
30=bad entity index
31=bad entity-parameter combination
32=bad parameter index
40=header bad number words
41=array size too large
(size of whole array and data returned)
42=Get LCN Time problems
44=duplicate entity name error

NOTE

Return Status codes 11 and 41 indicate warnings. Useful information has been returned. Return Status Code 5 indicates that the value status contains a non-normal data access status code.

Word 15—data type (integer) 1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
6=LCN external time (see 4.7.3)
7=real array
8=integer array
9=enumeration array
10=enum ordinal values array
13=internal entity id
14=internal entity id array
15=external entity id (ASCII name)
16=external entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array

Word 16—start of data record

Data record formats:

Real	Words 1 and 2—value (real) Word 3—value status (integer)
Integer	Word 1—value (integer) Word 2—value status (integer)
ASCII String (24 characters)	Words 1..12—value (ASCII characters) Word 13—value status (integer)
Enumeration	Words 1..4—value (ASCII characters) Word 5—value status (integer)
Ordinal	Word 1—ordinal value of enum (integer) Word 2—value status (integer)
Time	Words 1..9—time value (ASCII characters) Word 10—value status (integer)
Real Array	Word 1—array value status (integer) Words 2..3—array element 1 (real) Words 4..5—array element 2 (real) etc.
Integer Array	Word 1—array value status (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Enum Array	Word 1—array value status (integer) Words 2..5—array element 1 (ASCII char) Words 6..9—array element 2 (ASCII char) etc.

Ordinal Array	Word 1—array value status (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Internal Entity ID	Words 1..4—value (internal format) Word 5—value status (integer)
Internal Entity ID Array	Word 1—value status (integer) Words 2..5—array element 1 Words 6..9—array element 2 etc.
External Entity ID	Words 1..8—name (up to 16 ASCII char) Word 9—LCN id (2 characters; blank=local) Word 10—value status (integer)
External Entity ID Array	Word 1—value status (integer) Words 2..10—array element 1 Words 11..19—array element 2 etc.
LCN Internal Time	Words 1..2—seconds since 1 January 1979 (32-bit integer) Word 3—tenths of ms in current second (integer) Word 4—value status (integer)
LCN Internal Time Array	Word 1—value status (integer) Words 2..4—array element 1 Words 5..7—array element 2 etc.
ASCII String (40 characters)	Words 1..20—value (ASCII characters) Word 21—value status (integer)
ASCII String (40 characters) Array	Word 1—value status (integer) Words 2..21—array element 1 Words 22..41—array element 2 etc.

Use information: When data type is an array, the value status applies to the whole array. For value status definitions, see “Data Access Status Codes” in the *Messages Directory*.

4.7.5 Store Single Point Request (External Identifier)

This request from the host processor stores single value (or an array of values) for a point and parameter specified in the ACP's calling sequence, using the external identifier.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 35
Word 2—Number of Words = 16 plus data record size
Word 5—CG Identifier = 2

Message format:
Words 1..4—ACIDP name (8 ASCII characters)
Word 5—data access priority (integer) 1=control
2=noncontrol
Word 6—data type (integer) 1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array
Words 7..10—point name (8 ASCII characters)
Word 11—point name index (integer)
Words 12..15—parameter name (8 ASCII characters)
Word 16—parameter array pointer/size (integer)
Word 17—start of data record

Data Record formats:

Real	Words 1 and 2—value (real) Word 3—store status (integer)
Integer	Word 1—value (integer) Word 2—0 (integer)
ASCII String (24 characters)	Words 1..12—value (ASCII characters) Word 13—store status (integer)
Enumeration	Words 1..4—value (ASCII characters) Word 5—0 (integer)
Ordinal	Word 1—ordinal value of enumeration (integer) Word 2—0 (integer)

Real Array	Word 1—array store status (integer) Words 2..3—array element 1 (real) Words 4..5—array element 2 (real) etc.
Integer Array	Word 1—0 (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Enum Array	Word 1—0 (integer) Words 2..5—array element 1 (ASCII characters) Words 6..9—array element 2 (ASCII characters) etc.
Ordinal Array	Word 1—0 (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Internal Entity ID	Words 1..4—value (internal format) Word 5—value status (integer)
Internal Entity ID Array	Word 1—value status (integer) Words 2..5—array element 1 Words 6..9—array element 2 etc.
LCN Internal Time	Words 1..2—seconds since 1 January 1979 (32-bit integer) Word 3—tenths of ms in current second (integer) Word 4—value status (integer)
LCN Internal Time Array	Word 1—value status (integer) Words 2..4—array element 1 Words 5..7—array element 2 etc.
ASCII String (40 characters)	Words 1..20—value (ASCII characters) Word 21—value status (integer)
ASCII String (40 characters) Array	Word 1—value status (integer) Words 2..21—array element 1 Words 22..41—array element 2 etc.

Use information: Store status (applies only to reals and ASCII)
0=store the data value(s) provided
1=store the bad value representation instead

When data type is an array, the store status applies to the whole array. The bad value representations are NaN for real values and question mark strings for ASCII.

Content of the field "parameter array pointer/size" depends on data-type specification. See paragraph 4.7.1 for an explanation of this field's use with single parameters and with parameter arrays.

This message requires the requesting ACP be connected to its ACIDP.

This message may be used by one ACP to turn on another ACP by setting the other's ACIDP PPS parameter to ON.

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 135
Word 2—Number of Words = 31 plus data record size
Word 5—CG Identifier = 2

Message format: Words 1..8—ACIDP name (16 ASCII characters)
Word 9—data access priority (integer) 1=control
2=noncontrol
Word 10—data type (integer) 1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array
Word 11—LCN PIN node ID (2 ASCII characters)
Words 12..19—point name (16 ASCII characters)
Word 20—point name index (integer)
Words 21..24—parameter name (8 ASCII characters)
Word 25—parameter index x (integer)
Word 26—parameter index y (future use)
Words 27..31—reserved for future use (fill with integer zeros)
Word 32—start of data record

Data Record formats:

Real	Words 1 and 2—value (real) Word 3—store status (integer)
Integer	Word 1—value (integer) Word 2—0 (integer)
ASCII String (24 characters)	Words 1..12—value (ASCII characters) Word 13—store status (integer)
Enumeration	Words 1..4—value (ASCII characters) Word 5—0 (integer)
Ordinal	Word 1—ordinal value of enumeration (integer) Word 2—0 (integer)
Real Array	Word 1—array store status (integer) Words 2..3—array element 1 (real) Words 4..5—array element 2 (real) etc.
Integer Array	Word 1—0 (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Enum Array	Word 1—0 (integer) Words 2..5—array element 1 (ASCII characters) Words 6..9—array element 2 (ASCII characters) etc.
Ordinal Array	Word 1—0 (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Internal Entity ID	Words 1..4—value (internal format) Word 5—value status (integer)
Internal Entity ID Array	Word 1—value status (integer) Words 2..5—array element 1 Words 6..9—array element 2 etc.
LCN Internal Time	Words 1..2—seconds since 1 January 1979 (32-bit integer) Word 3—tenths of ms in current second (integer) Word 4—value status (integer)

LCN Internal Time Array	Word 1—value status (integer) Words 2..4—array element 1 Words 5..7—array element 2 etc.
ASCII String (40 characters)	Words 1..20—value (ASCII characters) Word 21—value status (integer)
ASCII String (40 characters) Array	Word 1—value status (integer) Words 2..21—array element 1 Words 22..41—array element 2 etc.

Use information:

Store status (applies only to reals and ASCII)
0=store the data value(s) provided
1=store the bad value representation instead

When data type is an array, the store status applies to the whole array. The bad value representations are NaN for real values and question mark strings for ASCII.

Content of the field "parameter array pointer/size" depends on data-type specification. See paragraph 4.7.1 for an explanation of this field's use with single parameters and with parameter arrays.

This message requires the requesting ACP be connected to its ACIDP.

This message may be used by one ACP to turn on another ACP by setting the other's ACIDP PPS parameter to ON.

4.7.6 Store Single Point Return (External Identifier)

This message from the CG is the response to a Store Single Point request (external identifier). The data returned is value status only.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 36
Word 2—Number of Words = 7
Word 5—CG Identifier = 2

Message format: Words 1..4—ACIDP name (8 ASCII characters)
Word 5—return status (integer)

0=normal
5=check value status
6=entity not found
7=data access failure
(complete error logged in SMCC display)
8=access key incorrect
11=data type mismatch
(data type corrected, no data stored)
14=array size too small (size < 1)
15=incorrect execution state
16=ACIDP name invalid
20=ACP-ACIDP not connected
21=invalid option value
22=data type out of range
26=parameter not found
28=bad data access priority
29=bad data type
30=bad entity index
31=bad entity-parameter combination
32=bad parameter index
40=header bad number words
43=invalid store code
44=duplicate entity name error

NOTE

Return Status code 11 indicates a warning. Useful information (correct data type code) has been returned. Return Status code 5 indicates that the value status contains a non-normal data access status code.

Word 6—data type (integer)
1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=enum ordinal values array
13=internal entity id
14=internal entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII array (40-char. string)

Word 7—value status (integer), see "Data Access Status Codes" in the *Messages Directory*.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 136
Word 2—Number of words = 16
Word 5—CG Identifier = 2

Message format:
Words 1..8—ACIDP name (16 ASCII characters)
Words 9..13—reserved for future use
Word 14—return status (integer)
0=normal
5=check value status
6=entity not found
7=data access failure
(complete error logged in SMCC display)
8=access key incorrect
11=data type mismatch
(data type corrected, no data stored)
14=array size too small (size < 1)
15=incorrect execution state
16=ACIDP name invalid
20=ACP-ACIDP not connected
21=invalid option value
22=data type out of range
26=parameter not found
28=bad data access priority
29=bad data type
30=bad entity index
31=bad entity-parameter combination
32=bad parameter index
40=header bad number words
43=invalid store code
44=duplicate entity name error

NOTE

Return Status code 11 indicates a warning. Useful information (correct data type code) has been returned. Return Status code 5 indicates that the value status contains a non-normal data access status code.

Word 15—data type (integer) 1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=enum ordinal values array
13=internal entity id
14=internal entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII array (40-char. string)

Word 16—value status (integer), see "Data Access Status Codes" in the *Messages Directory*.

4.7.7 Convert Identifier Request

This message from the host processor requests the conversion of a point/parameter name from external form (ASCII characters) to internal form.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 41
Word 2—Number of Words = 16
Word 5—CG Identifier = 2

Message format:
Words 1..4—ACIDP name or blank (8 ASCII characters)
Word 5—data access priority (integer) 1=control
2=noncontrol
Word 6—data type (integer) 1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
15=external entity id (ASCII name)
16=external entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array
Words 7..10—point name (8 ASCII characters)
Word 11—point name index (integer)
Words 12..15—parameter name (8 ASCII characters)
Word 16—parameter array pointer/size (integer)

Use information:
The internal address value for a given point/parameter can change whenever additions or deletions are made to the LCN database.

Point name index must = zero

Content of the field "parameter array pointer/size" depends on data-type specification. See paragraph 4.7.1 for an explanation of this field's use with single parameters and with parameter arrays.

Sixteen-Character Tagname Form

Header information:

Word 1—Transaction Code = 141
Word 2—Number of Words = 31
Word 5—CG Identifier = 2

Message format:

Words 1..8—ACIDP name or blank (16 ASCII characters)
Word 9—data access priority (integer) 1=control
2=noncontrol
Word 10—data type (integer) 1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
15=external entity id (ASCII name)
16=external entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array
Word 11—LCN PIN node ID (2 ASCII characters)
Words 12..19—point name (16 ASCII characters)
Word 20—point name index (integer)
Words 21..24—parameter name (8 ASCII characters)
Word 25—parameter index x (integer)
Word 26—parameter index y (future use)
Words 27..31—reserved for future use (fill with integer zeros)

Use information:

The internal address value for a given point/parameter can change whenever additions or deletions are made to the LCN database.

Point name index must = zero

Content of the field "parameter array pointer/size" depends on data-type specification. See paragraph 4.7.1 for an explanation of this field's use with single parameters and with parameter arrays.

4.7.8 Convert Identifier Return

This message from the CG is the response to a Convert Identifier request. The information returned is obtained by a search of LCN nodes.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 42
Word 2—Number of Words = 13
Word 5—CG Identifier = 2

Message format: Words 1..4—ACIDP name or blank (8 ASCII characters)
Word 5—return status (integer)

0=normal
5=check value status
6=entity not found
7=data access failure
(complete error logged in SMCC display)
11=data type mismatch
(data type corrected)
14=array size too small
(size of whole array returned)
15=incorrect execution state
16=ACIDP name invalid
20=ACP-ACIDP not connected
23=parameter index not zero
(index value corrected to zero)
26=parameter not found
(entity name converted)
27=parameter index out of range
28=bad data access priority
29=bad data type
30=bad entity index
31=bad entity-parameter combination
(only entity and parameter names are converted and good)
32=bad parameter index
40=header bad number words
41=array size too large
(size of whole array returned)
44=duplicate entity name error

NOTE

Return Status codes 11,14, 23, 26, 31, and 41 indicate warnings. Useful information has been returned.

Word 6—data type (integer)
1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
15=external entity id (ASCII name)
16=external entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array

Words 7..10—internal point identifier (integer array)
Word 11—internal parameter identifier (integer)
Word 12—parameter subscript (integer)
Word 13—enumeration set identifier (integer)

Use information:

When working with an array type, the parameter subscript contains actual array size, regardless of size specified in the request.

Message words 6-13 contain the CM50/CM60 "ID_Block."

Note that message word 11 (the internal parameter identifier) contains control bits that can vary from call to call to the same point.parameter. This variability has no significance in use of this 8-word block in Get or Store requests.

On a data type mismatch error (return status 11), the corrected data type is returned by the CG to the host in Word 6 Corrected enumeration/ordinal/boolean types are defaulted to type 4 or 9 (rather than to 5 or 10). Likewise, corrected internal/external entities are defaulted to types 13 or 14.

If the data type is truly a non-subscriptable element, but was specified by the host as an array (that is, the host incorrectly sends type 7-10, 14, 16, 18, or 20 instead of 1-5, 13, 15, 17, or 19), the CG returns the appropriate corrected element type. In all other cases the CG does not change the element/array code, but simply corrects the type of element or array.

Convert Identifier Request for array type parameters must specify on of the array types or, the data type in the Covert Identifier Return message will not be an array type. For Example:

Actual Data Type	Data Type in Covert ID Request	Data Type in Covert ID Response
real array	enumeration	real
real array	enumeration array	real array

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 142
Word 2—Number of Words = 22
Word 5—CG Identifier = 2

Message format: Words 1..8—ACIDP name or blank (16 ASCII characters)
Words 9..13—reserved for future use
Word 14—return status (integer)

0=normal
5=check value status
6=entity not found
7=data access failure
(complete error logged in SMCC display)
11=data type mismatch
(data type corrected)
14=array size too small
(size of whole array returned)
15=incorrect execution state
16=ACIDP name invalid
20=ACP-ACIDP not connected
23=parameter index not zero
(index value corrected to zero)
26=parameter not found
(entity name converted)
27=parameter index out of range
28=bad data access priority
29=bad data type
30=bad entity index
31=bad entity-parameter combination
(only entity and parameter names are converted and good)
32=bad parameter index
40=header bad number words
41=array size too large
(size of whole array returned)
44=duplicate entity name error

NOTE

Return Status codes 11,14, 23, 26, 31, and 41 indicate warnings. Useful information has been returned.

Word 15—data type (integer) 1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
15=external entity id (ASCII name)
16=external entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array

Words 16..19—internal point identifier (integer array)
Word 20—internal parameter identifier (integer)
Word 21—parameter subscript (integer)
Word 22—enumeration set identifier (integer)

Use information:

When working with an array type, the parameter subscript contains actual array size, regardless of size specified in the request.

Message words 6-13 contain the CM50/CM60 "ID_Block."

Note that message word 11 (the internal parameter identifier) contains control bits that can vary from call to call to the same point.parameter. This variability has no significance in use of this 8-word block in Get or Store requests.

On a data type mismatch error (return status 11), the corrected data type is returned by the CG to the host in Word 15. Corrected enumeration/ordinal/boolean types are defaulted to type 4 or 9 (rather than to 5 or 10). Likewise, corrected internal/external entities are defaulted to types 13 or 14.

If the data type is truly a non-subscriptable element, but was specified by the host as an array (that is, the host incorrectly sends type 7-10, 14, 16, 18, or 20 instead of 1-5, 13, 15, 17, or 19), the CG returns the appropriate corrected element type. In all other cases the CG does not change the element/array code, but simply corrects the type of element or array.

Convert Identifier Request for array type parameters must specify on of the array types or, the data type in the Covert Identifier Return message will not be an array type. For Example:

Actual Data Type	Data Type in Covert ID Request	Data Type in Covert ID Response
real array	enumeration	real
real array	enumeration array	real array

4.7.9 Get Single Point Request (Internal Address)

This request from the host processor is used to fetch a single value (or an array of values) for a point and parameter specified in the ACP's calling sequence, using the internal address.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 39
Word 2—Number of Words = 13
Word 5—CG Identifier = 2

Message format:
Words 1..4—ACIDP name or blank (8 ASCII characters)
Word 5—data access priority (integer) 1=control
2=noncontrol
Word 6—data type (integer) 1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
15=external entity id (ASCII name)
16=external entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array
Words 7..10—internal parameter identifier (integer array)
Word 11—internal parameter subscript (integer)
Word 12—parameter qualifier (integer)
Word 13—enumeration set identifier (integer)

Use information:
Words 6 through 10, 12, and 13 must be the same as previously obtained from the Convert Identifier Return (4.7.8). Word 11 may be changed depending on data–type specification. See paragraph 4.7.1 for an explanation of this field's use with single parameters and with parameter arrays.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 139
Word 2—Number of Words = 22
Word 5—CG Identifier = 2

Message format:
Words 1..8—ACIDP name or blank (16 ASCII characters)
Words 9..13—reserved for future use (fill with integer zeros)
Word 14—data access priority (integer) 1=control
2=noncontrol
Word 15—data type (integer) 1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
15=external entity id (ASCII name)
16=external entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array
Words 16..19—internal parameter identifier (integer array)
Word 20—internal parameter subscript (integer)
Word 21—parameter qualifier (integer)
Word 22—enumeration set identifier (integer)

Use information:
Words 15 through 19, 21, and 22 must be the same as previously obtained from the Convert Identifier Return (4.7.8.). Word 20 may be changed depending on data–type specification. See paragraph 4.7.1 for an explanation on this field's use with single parameters and with parameter arrays.

4.7.10 Get Single Point Return (Internal Address)

This message from the CG is the response to a Get Single Point request (internal address). The data returned includes both value and value status.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 40
Word 2—Number of Words = 6 plus size of data record
Word 5—CG Identifier = 2

Message format: Words 1..4—ACIDP name or blank (8 ASCII characters)
Word 5—return status (integer)

- 0=normal
- 5=check value status
- 7=data access failure (complete error logged in SMCC display)
- 11=data type mismatch (data type corrected, no data returned)
- 15=incorrect execution state
- 16=ACIDP name invalid
- 20=ACP-ACIDP not connected
- 22=data type out of range
- 27=parameter index out of range
- 28=bad data access priority
- 29=bad data type
- 30=bad entity index
- 31=bad entity-parameter combination
- 32=bad parameter index
- 40=header bad number words
- 41=array size too large (size of whole array and data returned)
- 44=duplicate entity name error

NOTE

Return Status codes 11 and 41 indicate warnings. Useful information has been returned. Return Status code 5 indicates that the value status contains a non-normal data access status code.

Word 6—data type (integer)

- 1=real
- 2=integer
- 3=ASCII (24-character string)
- 4=enumeration/Boolean
- 5=enumeration ordinal value
- 7=real array
- 8=integer array
- 9=enumeration array
- 10=ordinal values array

13=internal entity id
14=internal entity id array
15=external entity id (ASCII name)
16=external entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII array (40-char string)

Word 7—start of data record

Data record formats:

Real	Words 1 and 2—value (real) Word 3—value status (integer)
Integer	Word 1—value (integer) Word 2—value status (integer)
ASCII String (24 characters)	Words 1..12—value (ASCII characters) Word 13—value status (integer)
Enumeration	Words 1..4 —value (ASCII characters) Word 5—value status (integer)
Ordinal	Word 1—ordinal value of enumeration (integer) Word 2—value status (integer)
Real Array	Word 1—array value status (integer) Words 2..3—array element 1 (real) Words 4..5—array element 2 (real) etc.
Integer Array	Word 1—array value status (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Enum Array	Word 1—array value status (integer) Words 2..5—array element 1 (ASCII characters) Words 6..9—array element 2 (ASCII characters) etc.
Ordinal Array	Word 1—array value status (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Internal Entity ID	Words 1..4—value (internal format) Word 5—value status (integer)
Internal Entity ID Array	Word 1—value status (integer) Words 2..5—array element 1 Words 6..9—array element 2 etc.

External Entity ID	Words 1..8—name (up to 16 ASCII char) Word 9—LCN id (2 characters; blank=local) Word 10—value status (integer)
External Entity ID Array	Word 1—value status (integer) Words 2..10—array element 1 Words 11..19—array element 2 etc.
LCN Internal Time	Words 1..2—seconds since 1 January 1979 (32-bit integer) Word 3—tenths of ms in current second (integer) Word 4—value status (integer)
LCN Internal Time Array	Word 1—value status (integer) Words 2..4—array element 1 Words 5..7—array element 2 etc.
ASCII String (40 characters)	Words 1..20—value (ASCII characters) Word 21—value status (integer)
ASCII String (40 characters) Array	Word 1—value status (integer) Words 2..21—array element 1 Words 22..41—array element 2 etc.

Use information: When data type is an array, the value status applies to the whole array. For value status definitions see "Data Access Status Codes" in the *Messages Directory*.

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 140
Word 2—Number of Words = 15 plus size of data record
Word 5—CG Identifier = 2

Message format: Words 1..8—ACIDP name or blank (16 ASCII characters)
Words 9..13—reserved for future use
Word 14—return status (integer)

- 0=normal
- 5=check value status
- 7=data access failure (complete error logged in SMCC display)
- 11=data type mismatch (data type corrected, no data returned)
- 15=incorrect execution state
- 16=ACIDP name invalid
- 20=ACP-ACIDP not connected
- 22=data type out of range
- 27=parameter index out of range
- 28=bad data access priority
- 29=bad data type
- 30=bad entity index
- 31=bad entity-parameter combination
- 32=bad parameter index
- 40=header bad number words
- 41=array size too large (size of whole array and data returned)
- 44=duplicate entity name error

NOTE

Return Status codes 11 and 41 indicate warnings. Useful information has been returned. Return Status code 5 indicates that the value status contains a non-normal data access status code.

Word 15—data type (integer)

- 1=real
- 2=integer
- 3=ASCII (24-character string)
- 4=enumeration/Boolean
- 5=enumeration ordinal value
- 7=real array
- 8=integer array
- 9=enumeration array
- 10=ordinal values array
- 13=internal entity id
- 14=internal entity id array
- 15=external entity id (ASCII name)

16=external entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII array (40-char. string)

Word 16—start of data record

Data record formats:

Real	Words 1 and 2—value (real) Word 3—value status (integer)
Integer	Word 1—value (integer) Word 2—value status (integer)
ASCII String (24 characters)	Words 1..12—value (ASCII characters) Word 13—value status (integer)
Enumeration	Words 1..4 —value (ASCII characters) Word 5—value status (integer)
Ordinal	Word 1—ordinal value of enumeration (integer) Word 2—value status (integer)
Real Array	Word 1—array value status (integer) Words 2..3—array element 1 (real) Words 4..5—array element 2 (real) etc.
Integer Array	Word 1—array value status (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Enum Array	Word 1—array value status (integer) Words 2..5—array element 1 (ASCII characters) Words 6..9—array element 2 (ASCII characters) etc.
Ordinal Array	Word 1—array value status (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Internal Entity ID	Words 1..4—value (internal format) Word 5—value status (integer)
Internal Entity ID Array	Word 1—value status (integer) Words 2..5—array element 1 Words 6..9—array element 2 etc.
External Entity ID	Words 1..8—name (up to 16 ASCII char) Word 9—LCN id (2 characters; blank=local) Word 10—value status (integer)

External Entity ID Array	Word 1—value status (integer) Words 2..10—array element 1 Words 11..19—array element 2 etc.
LCN Internal Time	Words 1..2—seconds since 1 January 1979 (32-bit integer) Word 3—tenths of ms in current second (integer) Word 4—value status (integer)
LCN Internal Time Array	Word 1—value status (integer) Words 2..4—array element 1 Words 5..7—array element 2 etc.
ASCII String (40 characters)	Words 1..20—value (ASCII characters) Word 21—value status (integer)
ASCII String (40 characters) Array	Word 1—value status (integer) Words 2..21—array element 1 Words 22..41—array element 2 etc.

Use information: When data type is an array, the value status applies to the whole array. For value status definitions see "Data Access Status Codes" in the *Messages Directory*.

4.7.11 Store Single Point Request (Internal Address)

This request from the host processor stores a single value (or an array of values) for a specified point and parameter using its internal identification (address).

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 37
Word 2—Number of Words = 13 plus size of data record
Word 5—CG Identifier = 2

Message format:
Words 1..4—ACIDP name (8 ASCII characters)
Word 5—data access priority (integer) 1=control
2=noncontrol
Word 6—data type (integer) 1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array
Words 7..10—internal name identifier (integer array)
Word 11—internal parameter identifier (integer)
Word 12—parameter qualifier (integer)
Word 13—enumeration set identifier (integer)
Word 14—start of data record

Data record formats:

Real	Words 1 and 2—value (real) Word 3—store status (integer)
Integer	Word 1—value (integer) Word 2—0 (integer)
ASCII String (24 characters)	Words 1..12—value (ASCII characters) Word 13—store status (integer)
Enumeration	Words 1..4—value (ASCII characters) Word 5—0 (integer)

Ordinal	Word 1—ordinal value of enumeration (integer) Word 2—0 (integer)
Real Array	Word 1—array store status (integer) Words 2..3—array element 1 (real) Words 4..5—array element 2 (real) etc.
Integer Array	Word 1—0 (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Enum Array	Word 1—0 (integer) Words 2..5—array element 1 (ASCII characters) Words 6..9—array element 2 (ASCII characters) etc.
Ordinal Array	Word 1—0 (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Internal Entity ID	Words 1..4—value (internal format) Word 5—value status (integer)
Internal Entity ID Array	Word 1—value status (integer) Words 2..5—array element 1 Words 6..9—array element 2 etc.
LCN Internal Time	Words 1..2—seconds since 1 January 1979 (32-bit integer) Word 3—tenths of ms in current second (integer) Word 4—value status (integer)
LCN Internal Time Array	Word 1—value status (integer) Words 2..4—array element 1 Words 5..7—array element 2 etc.
ASCII String (40 characters)	Words 1..20—value (ASCII characters) Word 21—value status (integer)
ASCII String (40 characters) Array	Word 1—value status (integer) Words 2..21—array element 1 Words 22..41—array element 2 etc.

Use information: Store status (applies only to reals and ASCII)
0=store the data value(s) provided
1=store the bad value representation instead

When data type is an array, the store status applies to the whole array. The bad value representations are NaN for real values and question mark strings for ASCII.

This message can be used by one ACP to turn on another ACP by setting the other's PPS parameter to ON.

Words 6 through 13 must be the same as previously obtained from a Convert Identifier Return (4.7.8).

This message requires the requesting ACP be connected to its ACIDP.

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 137
Word 2—Number of Words = 22 plus size of data record
Word 5—CG Identifier = 2

Message format: Words 1..8—ACIDP name (16 ASCII characters)
Words 9..13—reserved for future use (fill with integer zeros)
Word 14—data access priority (integer) 1=control
2=noncontrol
Word 15—data type (integer) 1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array
Words 16..19—internal name identifier (integer array)
Word 20—internal parameter identifier (integer)
Word 21—parameter qualifier (integer)
Word 22—enumeration set identifier (integer)
Word 23—start of data record

Data record formats:

Real	Words 1 and 2—value (real) Word 3—store status (integer)
Integer	Word 1—value (integer) Word 2—0 (integer)
ASCII String (24 characters)	Words 1..12—value (ASCII characters) Word 13—store status (integer)
Enumeration	Words 1..4—value (ASCII characters) Word 5—0 (integer)
Ordinal	Word 1—ordinal value of enumeration (integer) Word 2—0 (integer)
Real Array	Word 1—array store status (integer) Words 2..3—array element 1 (real) Words 4..5—array element 2 (real) etc.
Integer Array	Word 1—0 (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Enum Array	Word 1—0 (integer) Words 2..5—array element 1 (ASCII characters) Words 6..9—array element 2 (ASCII characters) etc.
Ordinal Array	Word 1—0 (integer) Word 2—array element 1 (integer) Word 3—array element 2 (integer) etc.
Internal Entity ID	Words 1..4—value (internal format) Word 5—value status (integer)
Internal Entity ID Array	Word 1—value status (integer) Words 2..5—array element 1 Words 6..9—array element 2 etc.
LCN Internal Time	Words 1..2—seconds since 1 January 1979 (32-bit integer) Word 3—tenths of ms in current second (integer) Word 4—value status (integer)

LCN Internal Time Array	Word 1—value status (integer) Words 2..4—array element 1 Words 5..7—array element 2 etc.
ASCII String (40 characters)	Words 1..20—value (ASCII characters) Word 21—value status (integer)
ASCII String (40 characters) Array	Word 1—value status (integer) Words 2..21—array element 1 Words 22..41—array element 2 etc.

Use information:

Store status (applies only to reals and ASCII)
0=store the data value(s) provided
1=store the bad value representation instead

When data type is an array, the store status applies to the whole array. The bad value representations are NaN for real values and question mark strings for ASCII.

This message can be used by one ACP to turn on another ACP by setting the other's PPS parameter to ON.

Words 16 through 22 must be the same as previously obtained from a Convert Identifier Return (4.7.8).

This message requires the requesting ACP be connected to its ACIDP.

4.7.12 Store Single Point Return (Internal Address)

This message from the CG is the response to a Store Single Point request (internal address). The data returned is value status only.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 38
Word 2—Number of Words = 7
Word 5—CG Identifier = 2

Message format: Words 1..4—ACIDP name (8 ASCII characters)
Word 5—return status (integer)

- 0=normal
- 5=check value status
- 7=data access failure (complete error logged in SMCC display)
- 8=access key incorrect
- 11=data type mismatch (data type corrected, no data stored)
- 14=array size too small (size < 1)
- 15=incorrect execution state
- 16=ACIDP name invalid
- 20=ACP-ACIDP not connected
- 21=invalid option value
- 22=data type out of range
- 28=bad data access priority
- 29=bad data type
- 30=bad entity index
- 32=bad parameter index
- 40=header bad number words
- 43=invalid store code
- 44=duplicate entity name error

NOTE

Return Status code 11 indicates a warning. Useful information (correct data type code) has been returned. Return Status code 5 indicates that the value status contains a non-normal data access status code.

Word 6—data type (integer)

- 1=real
- 2=integer
- 3=ASCII (24-character string)
- 4=enumeration/Boolean
- 5=enumeration ordinal value
- 7=real array
- 8=integer array
- 9=enumeration array
- 10=ordinal values array

13=internal entity id
14=internal entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array

Word 7—value status (integer), see "Data Access Status Codes" in the *Messages Directory*.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 138
Word 2—Number of Words = 16
Word 5—CG Identifier = 2

Message format:
Words 1..8—ACIDP name (16 ASCII characters)
Words 9..13—reserved for future use
Word 14—return status (integer)

0=normal
5=check value status
7=data access failure
(complete error logged in SMCC display)
8=access key incorrect
11=data type mismatch
(data type corrected, no data stored)
14=array size too small (size < 1)
15=incorrect execution state
16=ACIDP name invalid
20=ACP-ACIDP not connected
21=invalid option value
22=data type out of range
28=bad data access priority
29=bad data type
30=bad entity index
32=bad parameter index
40=header bad number words
43=invalid store code
44=duplicate entity name error

NOTE

Return Status code 11 indicates a warning. Useful information (correct data type code) has been returned. Return Status code 5 indicates that the value status contains a non-normal data access status code.

Word 15—data type (integer) 1=real
2=integer
3=ASCII (24-character string)
4=enumeration/Boolean
5=enumeration ordinal value
7=real array
8=integer array
9=enumeration array
10=ordinal values array
13=internal entity id
14=internal entity id array
17=LCN internal time
18=LCN internal time array
19=ASCII (40-character string)
20=ASCII (40-char. string) array

Word 16—value status (integer), see “Data Access Status Codes” in the *Messages Directory*.

4.8 HISTORY DATA TRANSFERS

History data requests are limited to the data types that can be historized. These are primarily analog points with parameters that are expressed as real values; however, history snapshots can include digital points that are expressed as integer (ordinal) values.

Up-to-four concurrent History requests (from separate ACPs) can be accepted by the CG. A fifth request will be rejected with a queue-full return status.

4.8.1 Get History Request

This message is sent by the host processor to request the transmission of previously historized data from the HM.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 50
Word 2—Number of Words = 30 plus size of the Internal DDT table
Word 5—CG Identifier = 2

Message format:
Words 1..4—ACIDP name or blank (8 ASCII characters)
Word 5—history type (integer)
0=snapshots at 60-second interval
1=hourly averages
2=shift averages
3=daily averages
4=monthly averages
5=user averages
6=reserved for future use
7=monthly averages at fast interval
8=sampling rate enquiry
9=snapshots at 5-second interval
10=snapshots at 10-second interval
11=snapshots at 20-second interval
Word 6—history mode (integer) 0=absolute
1=relative
Words 7..17—beginning date and time (22 ASCII characters)
Words 18..28—ending date and time (22 ASCII characters)
Word 29—beginning offset (integer)
Word 30—ending offset (integer)
Word 31—start of Internal Data Table

Use information:
Same as for the sixteen-character tagname form explained below.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 150
Word 2—Number of Words = 39 plus size of the Internal DDT table
Word 5—CG Identifier = 2

Message format:
Words 1..8—ACIDP name or blank (16 ASCII characters)
Words 9..13—reserved for future use (fill with integer zeros)
Word 14—history type (integer)
0=snapshots at 60-second interval
1=hourly averages
2=shift averages
3=daily averages
4=monthly averages
5=user averages
6=reserved for future use
7=monthly averages at fast interval
8=sampling rate enquiry
9=snapshots at 5-second interval
10=snapshots at 10-second interval
11=snapshots at 20-second interval
Word 15—history mode (integer) 0=absolute
1=relative
Words 16..26—beginning date and time (22 ASCII characters)
Words 27..37—ending date and time (22 ASCII characters)
Word 38—beginning offset (integer)
Word 39—ending offset (integer)
Word 40—start of Internal Data Table

Use information:
Common information—The data table follows the Internal DDT format as defined in paragraph 4.5. The number of point/parameters is limited to 24. (Because of message size limitations, the maximum number of point parameters in a request for one hour of snapshots is 19.) See paragraph 4.8.2 for other restrictions on the amount of data that can be returned to a history data request.

The direction of search can be either forward (oldest to newest data) or backwards (newest to oldest data); however, a forward search requires at least twice as long to execute. To execute a backward search using offset values, set the starting offset value less than or equal to the ending offset value.

There is a limit of 262 samples per call.

Absolute requests—Beginning and ending date and time apply to only absolute history requests. For relative history requests, these fields should be blank (spaces).

Date and time format used in absolute history requests is MM/DD/YY HH:MM with eight trailing blank characters.

Relative requests—Beginning and ending offset times apply to only relative history requests. They are stated in the units of the specified history type and are used to establish start time, search direction, and the number of samples to be collected from the HM. Further explanation of the use characteristics of offset times follows:

Offset Units used for each of the history types are

History Type	Units of Offset
5-second snapshots	5 seconds
10-second snapshots	10 seconds
20-second snapshots	20 seconds
60-second snapshots	Minutes
Hourly averages	Hours
Shift averages	8-hour shift is assumed
Daily averages	Days
Monthly averages (normal or fast interval)	Months
User averages	Number of user averages to skip

The number of samples returned when offset values are used is calculated as the positive difference between the starting offset and the ending offset plus one. If this difference exceeds 262, the request is truncated at 262 samples and the return status is set to five (complete with errors).

The 5-, 10-, or 20-second snapshots are synchronized with the 60-second (1-minute) snapshots and samples returned are relative to the start of a minute.

Offset values less than one have special meanings. When the starting or ending offset value is zero (i.e., current LCN time), in the case of averages, the first sample returned is the current running average for the period.

A starting offset of -1 has special meaning in the cases of snapshots and user averages. In only those cases, LCN time is rounded to the beginning of the last hour. This permits an ACP to be sure of obtaining the last full hour of snapshots or user averages. For calculating the number of samples returned, a -1 is treated as a value of 0 and its number of samples and direction of search follows those rules.

An ending offset of -1 for snapshots and user averages means the search direction is forward and the ending time is on the hour starting "n" units back from current time.

For history type 8 (the sampling rate inquiry), both the starting and ending offsets should be set to zero.

The following table summarizes results of some possible combinations of starting and ending offsets with numbers of samples returned and reasons for zero-sample returns.

History Type	Starting Offset	Ending Offset	Number of Samples	Direction of Search	Partial First Sample?
any	0	0	1	Backward	yes
any	1	1	1	Backward	no
any	2	3	2	Backward	no
any	3	2	2	Forward	no
any	0	300	262	Backward	yes
0,5, 9-11	3	-1	4	Forward	no
1 to 4, 7	3	-1	0	Error, end offset	invalid
0,5, 9-11	-1	3	4	Backward	no
0,5, 9-11	-1	-3	0	Error, end offset	invalid
1 to 4	-1	-3	0	Error, begin/end offset	invalid
0, 9-11	-1	-1	1	Backward	--

4.8.2 Get History Return

This message is sent by the CG in response to a request for transmission of history data stored at the HM.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 51
Word 2—Number of Words = 5 plus size of the data table
Word 5—CG Identifier = 2

Message format: Words 1..4—ACIDP name or blank (8 ASCII characters)
Word 5—return status (integer)

- 0=normal return
- 4=begin period is invalid
- 5=complete with errors
- 6=end period is invalid
- 7=history type invalid or data access failed
- 9=HM rejects the request for history
- 10=the DDT was empty
- 13=begin-end periods are invalid
- 14=the requested number of samples exceed the size of the return buffer
- 15=ACIDP not in the correct EXECSTAT
- 16=ACIDP name is invalid
- 33=data type requested in the DDT is invalid
- 35=referenced DDT is not of type "history"
- 37=referenced DDT has duplicate data types
- 39=referenced DDT contains too many point.parameters
- 40=number of words in the request header is incorrect

Word 6—start of history data values

Data format: Same as for the sixteen-character tagname form explained below.

Use information: Same as for the sixteen-character tagname form explained below.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 151
Word 2—Number of Words = 14 plus size of the data table
Word 5—CG Identifier = 2

Message format:
Words 1..8—ACIDP name or blank (16 ASCII characters)
Words 9..13—reserved for future use
Word 14—return status (integer)
0=normal return
4=begin period is invalid
5=complete with errors
6=end period is invalid
7=history type invalid or data access failed
9=HM rejects the request for history
10=the DDT was empty
13=begin-end periods are invalid
14=the requested number of samples exceed the size of the return buffer
15=ACIDP not in the correct EXECSTAT
16=ACIDP name is invalid
33=data type requested in the DDT is invalid
35=referenced DDT is not of type "history"
37=referenced DDT has duplicate data types
39=referenced DDT contains too many point.parameters
40=number of words in the request header is incorrect
Word 15—start of history data values

Data format:
Each point/parameter's values for snapshots and averages are contained within variable length records that depend on history type. Each record begins with a 2-word record header, followed by a number of subrecords of either 5, 10, or 11 words each.

The record header format is

Word 1	data access status (integer)—See the *Messages Directory*
Word 2	number of subrecords for this point/parameter (integer)

If the Data Access Status Code is not 4, 5, 6 or 8, the number of subrecords is zero. When the number of subrecords is zero, only the header words are used.

For **Snapshots** (history type = 0, 9, 10, or 11), each subrecord is five words long, and conforms to one of three forms. Content of each subrecord is established by its value type (word 1 of each subrecord). See paragraph 4.8.2.1.

Form 1

Word 1	=0 (integer)—real process data value
Words 2..3	process value (real)
Words 4..5	timestamp (unsigned 32-bit integer)

Form 2

Word 1	=2 (integer)—digital process data value
Word 2	ordinal value of a self-defined enumeration (integer)
Word 3	not used (real, initialized to IEEE NaN)
Words 4..5	timestamp (unsigned 32-bit integer)

Form 3

Word 1	=5 (integer)—time change =6 (integer)—outage =7 (integer)—no data =11 (integer)—collection inhibited =12 (integer)—not in history =14 (integer)—fast collection not configured
Words 2..3	not used (real, initialized to IEEE NaN)
Words 4..5	timestamp (unsigned 32-bit integer)

For **Averages** (history type = 1,2,3,4, or 5), each subrecord is 10 words long, and conforms to one of two forms. Content of each subrecord is established by its value type (word 1 of the subrecord). See paragraph 4.8.2.2.

Form 1

Word 1	=0 (integer)—normal data, or =1 (integer)—nonstandard, or =5 (integer)—time change, or =13 (integer)—time change nonstandard
Words 2..3	process value (real)
Words 4..5	timestamp (unsigned 32-bit integer)
Words 6..7	maximum process value* in period (real)
Words 8..9	minimum process value* in period (real)
Word 10	number of samples (unsigned 16-bit integer)

Form 2

Word 1	=6 (integer)—outage, or =7 (integer)—no data, or =12 (integer) not in history
Words 2..3	not used (real, initialized to IEEE NaN)
Words 4..5	timestamp (unsigned 32-bit integer)
Words 6..7	not used (real, initialized to IEEE NaN)
Words 8..9	not used (real, initialized to IEEE NaN)
Word 10	not used

*Because of the storage method used, Minimum/Maximum process values can have up to 1% error (+1% for maximums and -1% for minimums).

For **Fast Interval Monthly Averages** (history type = 7) derived from 5-, 10-, or 20-second snapshots, each subrecord is 11 words long.

Form 1

Word 1	=0 (integer)—normal data, or =1 (integer)—nonstandard, or =5 (integer)—time change, or =13 (integer)—time change nonstandard
Words 2..3	process value (real)
Words 4..5	timestamp (unsigned 32-bit integer)
Words 6..7	maximum process value* in period (real)
Words 8..9	minimum process value* in period (real)
Word 10..11	number of samples (32-bit integer)

Form 2

Word 1	=6 (integer)—outage, or =7 (integer)—no data, or =12 (integer) not in history
Words 2..3	not used (real, initialized to IEEE NaN)
Words 4..5	timestamp (unsigned 32-bit integer)
Words 6..7	not used (real, initialized to IEEE NaN)
Words 8..9	not used (real, initialized to IEEE NaN)
Word 10..11	number of samples skipped (32-bit integer)

For a **Sampling Rate Inquiry** (history type = 8), two data words are returned for each point.parameter defined in the DDT. Note that no record headers are included with this data.

Word 1	data access status (integer)—See the *Messages Directory*.
Word 2	sampling rate in seconds (integer)

Use information:

There are two limits placed on the amount of history data that can be returned for each request:

1) There can be no more than 262 data samples for any one point.

2) The maximum return-message size is 5986 data words.

The return message's contents are truncated at the point where either of these limits is reached.

The "timestamp" is a count of seconds since January 1, 1979

Digital process values included in Get History returns are reported as the ordinal values of user-defined enumerations.

*Because of the storage method used, Minimum/Maximum process values can have up to 1% error (+1% for maximums and -1% for minimums).

When snapshot data (history type 0, 9, 10, or 11) is requested, all the data returned is controlled by the time interval specified in the request (60-, 5-, 10, or 20-second intervals, respectively).

This means that, for example, if a 5-second interval is requested, and some of the points requested are in 60-second history groups, then each of the 60-second points has 11 "fast collection not configured" records (each with the appropriate 5-second timestamp) returned between the 60-second time values that are available. Any points in 10- or 20-second history groups are handled similarly. The following illustration shows when data records for each history group can be returned when a 5-second interval is requested. (Blank elements in the illustration indicate when a "fast collection not configured" record is returned instead of a data record.)

Timestamp	5-Second Group	10-Second Group	20-Second Group	60-Second Group
00:00:00	data record	data record	data record	data record
00:00:05	data record			
00:00:10	data record	data record		
00:00:15	data record			
00:00:20	data record	data record	data record	
00:00:25	data record			
00:00:30	data record	data record		
00:00:35	data record			
00:00:40	data record	data record	data record	
00:00:45	data record			
00:00:50	data record	data record		
00:00:55	data record			
00:01:00	data record	data record	data record	data record

4.8.2.1 Value-Type Meanings for Snapshots

0 = Normal Data: Value returned is analog (real) data.
1 = Nonstandard: not applicable
2 = Digital Value: Value returned is a self-defined enumeration.
3-4 = not used
5 = Time Change: A time change occurred and data for one minute is missing; value field contains NaN
6 = Outage: History module was not in service; value field contains NaN.
7 = No Data: The Data Owner was not in service; value field contains NaN.
8-10 = not used
11 = Collection Inhibited: History collection was not enabled; value field contains NaN.
12 = Not in History: Requested data was outside span of the History file; value field contains NaN.
13 = Time Change nonstandard: not applicable
14 = Fast Collection is not configured; value field contains NaN.

4.8.2.2 Value-Type Meanings for Averages

Averages are of analog data and are the total of the summed good averages for the period divided by the number of good samples.

0	=	Normal: 90% or more good samples
1	=	Nonstandard: Less than 90% good samples
2	=	Digital Value: not applicable (if an average is requested for a parameter of type digital, the value type returned is "data not in history").
3-4	=	not used
5	=	Time Change: A time change occurred during the averaging period, but there are 90% or more good samples.
6	=	Outage: History module was not in service for the entire period; value field contains NaN.
7	=	No Data: No values were available from the Data Owner for entire period; value field contains NaN.
8-10	=	not used
11	=	Collection Inhibited: not applicable
12	=	Not in History: Requested data was outside span of the History file; value field contains NaN.
13	=	Time Change nonstandard: The average calculation was performed according to the new time. Samples already collected are rolled into the new average. There are fewer than 90% good samples.

4.8.2.3 Effect of Time Change on History Gathering

Absolute History Requests—If a time change occurs during an Absolute History interval, the number of samples returned can differ from the expected number of samples. For example, if it is desired to obtain a day's worth of hourly averages (24) and a forward time change occurred, 23 samples are returned. If the time change is in the backward direction, 25 samples are returned.

Relative History Requests—The number of samples returned by a Relative History request is immune to time changes. The number of samples returned is always the absolute difference between the start and end offsets plus one.

Snapshot History Requests—The 60-second history collection task executes at the beginning of each minute (:00 seconds). When a time change is detected, the operating system resynchronizes all cyclic tasks beginning with the first interval following the time change.

For example, assume that current time is changed from 13:37:23 to 14:37:00. In this instance, the 60-second history collection will have collected data at 13:37:00, and will be reactivated at 14:38:00 (the first interval following the time change). At this time, a time change record is stored into the history collection files in lieu of a collected value. Starting at the next activation, at 14:39:00, history data is again collected. Thus, one 60-second history collection is skipped.

The 5-, 10-, and 20-second history data collections—which are synchronized with the 60-second collection—also replace their first possible data collections following a time change with time change records.

As shown by Figure 4-1, the remainder of the history record for the minute when the time change occurred is filled with "no data" indicators in order for the four different collection rates to stay in alignment. Note that the slower collection intervals fill with "fast collect not configured" indicators on each 5-second interval when they are not collecting data. (In the illustration, "dnc " represents a "data not collected" record, and "fcnc" indicates a "fast collection not configured" record.)

Timestamp	5-Second Group	10-Second Group	20-Second Group	60-Second Group
13:37:00	data record	data record	data record	data record
13:37:05	data record	fcnc	fcnc	fcnc
13:37:10	data record	data record	fcnc	fcnc
13:37:15	data record	fcnc	fcnc	fcnc
13:37:20	data record	data record	data record	fcnc
13:37:25	dnc	fcnc	fcnc	fcnc
13:37:30	dnc	dnc	fcnc	fcnc
13:37:35	dnc	fcnc	fcnc	fcnc
13:37:40	dnc	dnc	dnc	fcnc
13:37:45	dnc	fcnc	fcnc	fcnc
13:37:50	dnc	dnc	fcnc	fcnc
13:37:55	dnc	fcnc	fcnc	fcnc
14:38:00	time change	time change	time change	time change
14:38:05	data record	fcnc	fcnc	fcnc
14:38:10	data record	data record	fcnc	fcnc
14:38:15	data record	fcnc	fcnc	fcnc
14:38:20	data record	data record	data record	fcnc
14:38:25	data record	fcnc	fcnc	fcnc
14:38:30	data record	data record	fcnc	fcnc
14:38:35	data record	fcnc	fcnc	fcnc
14:38:40	data record	data record	data record	fcnc
14:38:45	data record	fcnc	fcnc	fcnc
14:38:50	data record	data record	fcnc	fcnc
14:38:55	data record	fcnc	fcnc	fcnc
14:39:00	data record	data record	data record	data record

4.9 TEXT MESSAGE TRANSFERS

4.9.1 Get Message Request

This message is used by the host processor to request the fetching of a message placed in an ACIDP by a CL/MC sequence program in a Multifunction Controller or by a CL/PM sequence program in a Process Manager Module.

Eight-Character Tagname Form

Header information: Word 1—Transaction Code = 13
Word 2—Number of Words = 4
Word 5—CG Identifier = 3

Message format: Words 1..4—ACIDP name (8 ASCII characters)

Sixteen-Character Tagname Form

Header information: Word 1—Transaction Code = 113
Word 2—Number of Words = 13
Word 5—CG Identifier = 3

Message format: Words 1..8—ACIDP name (16 ASCII characters)
Words 9..13—reserved for future use (fill with integer zeros)

4.9.2 Get Message Return

This message is used by the CG to respond to a request for fetching of a message placed in an ACIDP by a CL/MC or CL/PM sequence program.

Header information: Word 1—Transaction Code = 14
Word 2—Number of Words = 2 plus size of the message
Word 5—CG Identifier = 3

Message format: Word 1—return status (integer)
0=normal return
1=message too large
2=no message sent
4=normal return, second message queued
5=block record out of bounds
6=valid ACIDP not found
15=incorrect EXECSTAT
Word 2—message size in characters (integer)
Word 3—start message (ASCII characters)

Use information: Maximum message size is 120 characters.

4.9.3 Send Message Request

This message is used by the host processor to request the sending of a message to the operator personality in all Universal Stations monitoring the Area containing the Unit to which the ACIDP is assigned by point configuration.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 15
Word 2—Number of Words = 7 plus message size
Word 5—CG Identifier = 3

Message format:
Words 1..4—ACIDP name (8 ASCII characters)
Word 5—message confirmation required (integer) 1=true 0=false
Word 6—wait time (integer, 0-3600)
Word 7—destination (integer) 0=CRT 1=printer 2=both
Word 8—start of message (ASCII text)

Use information: Same as for the sixteen-character tagname form explained below.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 115
Word 2—Number of Words = 16 plus message size
Word 5—CG Identifier = 3

Message format:
Words 1..8—ACIDP name (16 ASCII characters)
Word 9..13—reserved for future use (fill with integer zeros)
Word 14—message confirmation required (integer) 1=true 0=false
Word 15—wait time (integer, 0-3600)
Word 16—destination (integer) 0=CRT 1=printer 2=both
Word 17—start of message (ASCII text)

Use information: The "wait time" is the time in seconds that the Universal Station is to wait for an operator response. (Note that since the ACP Scheduler runs only once each five seconds, the actual wait time can be up to 5 seconds less than specified in this message.) This parameter is ignored if message confirmation is not required or the destination is printer-only.

Message-length maximums are dependent on destination: 60 characters for CRT messages, 72 characters for printing, and HM journaling (all messages to the operator are journaled if the History Module is so configured). Oversize messages are truncated.

Two types of Event Initiated reports can be invoked by specially formatted messages from an ACP to the Area Universal Station.

- Logs, reports, journals, and trends configured in the Area database
- Event History reports

Details of message requirements for these reports are given in Section 30 of the *Engineer's Reference Manual* located in the *Implementation/Startup & Reconfiguration - 2* binder .

4.9.4 Send Message Response-Immediate

This message is used by the CG as the immediate response to a send message request to confirm the message transmission on the LCN.

Header information: Word 1—Transaction Code = 16
Word 2—Number of Words = 1
Word 5—CG Identifier = 3

Message format: Word 1—return status (integer) 0=normal return
2=no message sent
6=valid ACIDP not found
15=incorrect EXECSTAT

4.9.5 Message Confirmation or Timeout

This message is sent by the CG as the final response to a send message requiring operator response. It indicates either operator action or timeout.

Eight-Character Tagname Form

Header information:
Word 1—Transaction Code = 1
Word 2—Number of Words = 10
Word 5—CG Identifier = 1

Message format:
Words 1..4 —ACIDP name (8 ASCII characters)
Word 5—OPER_DMD (integer)
Word 6—PPS (integer)
Word 7—TAKE_I_P (integer)
Word 8—SCHEDULED (integer)
Word 9—PS_MSG (integer)
Word 10—return status (integer) 0=normal
4=timeout

Use information:
Words 5 through 9 can contain scheduling information for the ACP connected to this ACIDP (0=false, 1=true). See paragraphs 4.4.6 and 6.2 for additional information.

Sixteen-Character Tagname Form

Header information:
Word 1—Transaction Code = 101
Word 2—Number of Words = 19
Word 5—CG Identifier = 1

Message format:
Words 1..8 —ACIDP name (16 ASCII characters)
Words 9..13—reserved for future use
Word 14—OPER_DMD (integer)
Word 15—PPS (integer)
Word 16—TAKE_I_P (integer)
Word 17—SCHEDULED (integer)
Word 18—PS_MSG (integer)
Word 19—return status (integer) 0=normal
4=timeout

Use information:
Words 5 through 9 can contain scheduling information for the ACP connected to this ACIDP (0=false, 1=true). See paragraphs 4.4.6 and 6.2 for additional information.

4.10 FILE TRANSFERS

4.10.1 Introduction to File Transfers

The following paragraphs explain the format and content of the file transfer requests that a host processor can make to an HM on the local LCN. Some of these file requests also can be made across the Network Gateway to an HM on a remote LCN.

The host initiates all file transfer operations and can also request abort of any operation. For each file transfer request from the host there is a paired response from the CG. Descriptions of each file transfer request/response pair begin at heading 4.10.2.

4.10.1.1 File Transfer Message Format

Each file transaction request or response includes two fixed-length blocks and may include an optional data block of variable length. The fixed-length blocks are

- An 8-word message header (see heading 4.1.2.1 for this format)
- A 40-word message area. Some fields in this message area are fixed, and are echoed back in the response. Other fields in the message area will contain response information from the HM.

The optional data block which follows the message block can be up to 5760 words long for read/write operations. Values in the individual file transfer transactions are 16-bit integer unless otherwise noted.

4.10.1.2 Transaction and Command Codes

The same transaction codes are used for all file transfer request/response pairs.

- File Transfer Request—Transaction code 62
- File Transfer Response—Transaction code 63

The specific file transfer command to be executed is specified in a command code located in word 9 of the message block. A summary of the file transfer commands follows.

Command Code	File Transfer Request	Section Heading
1	Read LCN File	4.10.2
2	Write LCN File	4.10.3
3	Retrieve Single File Attributes	4.10.4
4	List Catalog into a File	4.10.5
5	Retrieve Volumes/Directories for a Device	4.10.6
6	List Volumes into a File	4.10.7
7	Copy File	4.10.8
8	Move File	4.10.9
9	Rename File	4.10.10
10	Create Directory	4.10.11
11	Delete Directory	4.10.12
12	Delete Files	4.10.13
13	Break Request	4.10.14
14	Data Out Request	4.10.15
15	Retrieve File Names and Extensions	4.10.16

4.10.1.3 Status Codes Returned by File Transactions

The message block of each File Transfer response contains two words of status data.

- Word 11 of the message block contains the code specifying the overall return status (0 = normal). See heading 4.10.17.1 for details.
- Word 12 of the message block contains a substatus code that is significant only when word one contains the value 1 (see heading 4.10.17.2) or the value 14 (see heading 4.10.17.3). Exception: A normal response to a Data Out request (Command Code = 14) uses the substatus code value to identify the Data Out on/off status.

4.10.1.4 HM Node Addressing Forms

The pathname used to address a History Module file can specify either NET or the physical node (PN:nn) of the History Module. The physical node number must be used when the HM is running the HMI personality. The maximum pathname length is 28 characters. Examples of the two addressing forms are:

```
NET>vdir>file_id.extension
PN:nn>vdir>file_id.extension
```

where
nn = the HM physical node number
vdir = volume or directory identifier
file_id.extension = the file name and suffix

Note that the "PN:nn" form of addressing is required when the History Module's initialization (HMI) personality is loaded.

Some file operations can be performed on remote LCN networks. A remote network address uses a 3-character prefix to the pathname that identifies the remote network. (e.g., n1\PN:43>vdir>filename.yy) Remote network addressing is limited to use with the following file transfer commands only:

- Read LCN NET File
- Write LCN NET File
- Retrieve Single File Attributes
- List Catalog Into a File
- Copy File
- Delete Files
- Data Out Request
- Retrieve File Names and Extensions

NOTE

In the descriptions of the various request/response pairs that follow, an asterisk (*) indicates that data in the field is a file attribute.

4.10.1.5 Use of EP Utilities Options

Several of the File Transfer transactions make use of one or more command options associated with the Engineering Personality Utilities. These options are invoked by adding "-" followed by the option name to the filename. More than one option can be included and each is separated by a space. For example a filename showing three of the available EP options would have this form:

NET>vdir>*.* -D -REC -FD

The available EP options are:

- -D Effect of the Data Out Option varies with the type of transaction it is used with.

  - Use of the -D option with copy, remove, rename or delete transactions results in output of a copy of the command and the list of files it causes to be accessed to a text file specified by the Data Out "ON" request. This text file then can be read by the host through the Read File request.

  - Use of the -D option with a list catalog transaction results in transfer to the text file only of volume information such as maximum files, total sectors, sectors in use, etc.

- -REC (or -A) Results in the listing of volume information and the following components: file name, extension, file type, protection status, version number, number of records, record size, number of blocks, file block size, starting sector, ending sector, and timestamp.

- -FD Results in the listing of volume information and the following components: file name, extension, file type, protection status, version number, timestamp, and the 64-character file description.

- -BF Results in the listing of volume information and the following components: file name, extension, file type, and protection status.

- -D Results in the listing of volume information only (maximum files, total sectors, sectors in use, etc.).

Detailed information on the EP Utility Options can be found in the *Utility Operations* manual.

The EP Utilities options that can be invoked with specific File Transfer transactions are:

- List Catalog Into a File (-D, -REC, -FD)
- Copy File (-D)
- Move File (-D)
- Rename File (-D)
- Delete Files (-D)

4.10.2 Read LCN File

This request/response pair reads a single file and its attributes from a volume on an HM to the host. Contiguous or linked files can be transferred.

Record size is affected by file type.

- Contiguous files have a fixed record size of 256 bytes.
- Linked files can have either a fixed or a variable record size.
  - The record size common to all records of a fixed-length linked file is returned in word 34 of the response message.
  - The record size of each variable-length record is found in the first two bytes of the record itself.

The maximum file data size in a single transmittal is 5760 words, thus several transmission request/response pairs can be required to transfer a complete file. Partial files can be transmitted, but a record or block is always complete within one transmission.

Files being read from the HM are opened for shared read. Since read operations can hold up HM write operations, the recommended method when reading system files that are periodically written to is to first copy the HM file locally to a user volume. The copy is then read to the host. If a file write request is occurring on the LCN when a file read is requested from the host, a busy abort is returned.

The reading of files from continuous history volumes and directories is prohibited while the HM's online (HMO) personality is loaded. Reads are permitted to all volumes while the HM's initialization (HMI) personality is loaded.

4.10.2.1 Read LCN File Request

Header information:

Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:

Words 1..8—Reserved for future use
Word 9—Command type = 1
Word 10—File handling when file does not fit within maximum transfer size: 1 = return partial file
2 = return abort error
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the file (ASCII characters)
(PN:nn>vdir>filename.xx or NET>vdir>filename.xx)
Word 27—Number of records requested 0 = all
Word 28—Starting record requested
Word 29—Ending record requested
(Words 28 and 29 are not used when Word 27 = 0)
Words 30..40—Reserved for response data

4.10.2.2 Read LCN File Response

Header information:
Word 1—Transaction Code = 63
Word 2—Number of Words = 40 + data size
Word 5—CG Identifier = 2

Message format:
Words 1..10—Request echo
Word 11—Return status 0 = normal
Word 12—Return substatus 0 = normal
Words 13..29—Request echo
Word 30—Total number of records in file*
Word 31—Number of records in this transmission
Word 32—First record number of this transmission
Word 33—Last record number of this transmission
Word 34—Record size*
Word 35—File type* 0 = contiguous, 1 = linked
Word 36—File block size* 4 to 45 for linked file
0 for contiguous file
Word 37—Number of blocks used*
Word 38—File configuration* 0 to 511
Word 39—File revision* 0 to 63
Word 40—Write access code* 0 = unprotected

Data information:
Words 1..32—File descriptor* (ASCII characters)
Words 33..5792—File data

4.10.3 Write LCN File

This request/response pair writes a single file and its attributes from the host to a volume on the LCN's HM. Contiguous or linked files can be transferred.

Record size is affected by file type.

- Contiguous files have a fixed record size of 256 bytes.
- Linked files can have either a fixed or a variable record size.
  - The record size common to all records of a fixed-length linked file is specified in word 28 of the request message.
  - The record size of each variable-length record is found in the first two bytes of the record itself.

The maximum file data size in a single transmittal is 5760 words, thus several transmission request/response pairs can be required to transfer a complete file. Partial files can be transmitted, but a record is always complete within one transmission.

A temporary file with a unique name is created to receive the data on the requested volume. After the transmission is complete, the temporary file is Safe Renamed to the requested LCN pathname.

If the contents of an existing file are over written or data are appended to the file, the resulting file attributes of the file are those specified in the message block of the request.

4.10.3.1 Write LCN File Request

Header information:
Word 1—Transaction Code = 62
Word 2—Number of Words = 40 + data size
Word 5—CG Identifier = 2

Message format:
Words 1..8—Reserved for future use
Word 9—Command type = 2
Word 10—File handling code
1 = Replace existing file
2 = Return error if file exists
3 = Append to or amend existing file
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the file (ASCII characters)
(PN:nn>vdir>filename.xx or NET>vdir>filename.xx)
Word 27—File Block Size* if linked file
(0 = default of 8)
Word 28—Record size* if linked file
(0 = variable size)
Word 29—Number of records in this transmission
Word 30—Requested starting record
(0 = append/amend)
Word 31—Requested ending record
(not applicable if word 30 = 0)
Word 32—Number of blocks in file*
(applies to contiguous files only)
Word 33—Transmission counter
(initial = 0, subsequent = previous response value)
Word 34—Last data block flag
(0 = not last)
Word 35—File type
(0 = contiguous, 1 = linked)
Word 36—Reserved for future use
Word 37—Number of blocks used*
Word 38—File configuration*
Word 39—File Revision*
Word 40—Write access code* (0 = unprotected)

Data information:
Words 1..32—File descriptor* (ASCII characters)
Words 33..5792—File data

4.10.3.2 Write LCN File Response

Header information:
Word 1—Transaction Code = 63
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..10—Request echo
Word 11—Return status 0 = normal
Word 12—Return substatus 0 = normal
Words 13..32—Request echo
Word 33—Request transmission counter +1
Word 34..40—Request echo

4.10.4 Retrieve Single File Attributes

This request/response pair is used to fetch the file attributes of a specified file. The attributes include the file type, file descriptor, write access code, block size, number of blocks, record size, number of records, time stamp, user file configuration number, and user file revision number.

No wild card characters or options can be used in the LCN pathname field.

4.10.4.1 Retrieve Single File Attributes Request

Header information:
Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..8—Reserved for future use
Word 9—Command type = 3
Word 10—Not used
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the file (ASCII characters)
(PN:nn>vdir>filename.xx or NET>vdir>filename.xx)
Words 27..40—Reserved for response data

4.10.4.2 Retrieve Single File Attributes Response

Header information:
Word 1—Transaction Code = 63
Word 2—Number of Words = 76
Word 5—CG Identifier = 2

Message format:
Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus (0 = normal)
Words 13..26—Request echo
Word 27—Write access code (0 = unprotected)
Word 28—File Type (0 = contiguous, 1 = linked)
Word 29—File record size for fixed length linked files only
(otherwise = 0)
Words 30..31—EOF Number of blocks
Word 32—File block size for linked files only
(otherwise = 0)
Words 33..34—Directory time stamp
(seconds since 1/01/79 00:00)
Word 35—File configuration (0..511)
Word 36—File revision (0..63)
Words 37..38—EOF logical record number (for linked files)
Words 39..40—Reserved for future use

Data information:
Words 1..32—File descriptor* (ASCII characters)
Words 33..34—Starting sector on volume
Words 35..36—Ending sector on volume

4.10.5 List Catalog into a File

This request/response pair is used to list file attributes of a file or files into an ASCII text catalog file. A Data Out request must have been made previously to specify the pathname of the catalog file. The catalog file can be transferred to the host processor by the Read File request.

The catalog file content varies according to information requested.

Wild card characters can be used to specify the file(s) to be cataloged. The -FD, -REC, and -D options are supported.

4.10.5.1 List Catalog into a File Request

Header information:
Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..8—Reserved for future use
Word 9—Command type = 4
Word 10—Not used
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the device/volume/file to catalog (ASCII characters)
(PN:nn>vdir>*.* -rec -fd or NET>vdir>*.* -rec -fd)
Words 27..40—Reserved for response data

4.10.5.2 List Catalog into a File Response

Header information:
Word 1—Transaction Code = 63
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus (0 = normal)
Words 13..26—Request echo
Words 27..40—LCN pathname to the Data Out ASCII file

4.10.6 Retrieve Volumes/Directories for a Device

This request/response pair is used to fetch the volume names, directory names, and sector usage associated with the volumes of a specified HM. No wild card characters or options are applicable with this command.

4.10.6.1 Retrieve Volumes/Directories for a Device Request

Header information:
Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..8—Reserved for future use
Word 9—Command type = 5
Word 10—Not used
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the device (ASCII characters) (PN:nn)
Word 27—Reserved for response data
Words 28..40—Reserved for future use

4.10.6.2 Retrieve Volumes/Directories for a Device Response

Header information:
Word 1—Transaction Code = 63
Word 2—Number of Words = 40 + 4050
Word 5—CG Identifier = 2

Message format:
Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus (0 = normal)
Words 13..26—Request echo
Word 27—Total number of volumes (1..30)
Word 28..40—Reserved for future use

Data information:
Word 1—Total number of directories in volume 1
Words 2..3—Total number of sectors (32-bit integer)
Words 4..5—Sectors in use (32-bit integer)
Words 6..7—Reserved for future use
Words 8..9—Volume Id for volume 1 (4 ASCII characters)
Words 10..11—Directory Id 1 for volume 1 (4 ASCII characters)
Words 12..135—Directory Ids 2..63 for Volume 1
Words 136..4050—Volume and Directory Ids for Volumes 2..30

4.10.7 List Volumes into a File

This request/response pair is used to list the volume/directory names and attributes of the "NET" or of a selected physical HM node into an ASCII file. Wild Card characters and options are not applicable. A Data Out request must have been made previously to specify the pathname of the text file. The text file can be transferred to the host processor by the Read File request.

4.10.7.1 List Volumes into a File Request

Header information: Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format: Words 1..8—Reserved for future use
Word 9—Command type = 6
Word 10—Not used
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the device (ASCII characters PN:nn or NET)
Words 27..40—Reserved for response data

4.10.7.2 List Volumes into a File Response

Header information: Word 1—Transaction Code = 63
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format: Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus (0 = normal)
Words 13..26—Request echo
Words 27..40—LCN pathname of the Data Out file (ASCII characters)

4.10.8 Copy File

This request/response pair is used to copy an HM file and place the copy in a destination file on an HM. Wild card characters are permitted. The destination suffix must always be the same as the source suffix. The -D option outputs to the Data Out file if specified in the destination path.

4.10.8.1 Copy File Request

Header information: Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format: Words 1..8—Reserved for future use
Word 9—Command type = 7
Word 10—Not used
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the HM source file (PN:nn>vdir>filename.xx or NET>vdir>filename.xx or NET>vdir>*.*)
Words 27..40—LCN pathname for the HM destination file (PN:nn>vdir>filename -D or NET>vdir>filename -D or NET>vdir>= -D)

4.10.8.2 Copy File Response

Header information: Word 1—Transaction Code = 63
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format: Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus (0 = normal)
Words 13..40—Request echo

4.10.9 Move File

This request/response pair is used to move a file from one directory to another directory in the same HM volume. Wild cards are supported in file names to allow multiple or selective file movement. The -D option lists file names into the specified Data Out file.

4.10.9.1 Move File Request

Header information: Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format: Words 1..8—Reserved for future use
Word 9—Command type = 8
Word 10—Not used
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the HM source file
(NET>dir1>filename.xx)
Words 27..40—LCN pathname for the HM destination directory
(dir2 -D)

4.10.9.2 Move File Response

Header information: Word 1—Transaction Code = 63
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format: Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus (0 = normal)
Words 13..40—Request echo

4.10.10 Rename File

This request/response pair is used to give a new name to a file on the HM. The original name no longer will be recognized by the LCN system unless that name is given to some other file. A wild card can be used on the source extension to rename multiple files. The -D option lists the file names to the Data Out file if specified.

4.10.10.1 Rename File Request

Header information:
Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..8—Reserved for future use
Word 9—Command type = 9
Word 10—Not used
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the old file name
(NET>vdir>filename.xx)
Words 27..40—New filename plus option
(filename -D)

4.10.10.2 Rename File Response

Header information:
Word 1—Transaction Code = 63
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus (0 = normal)
Words 13..40—Request echo

4.10.11 Create Directory

This request/response pair is used to create a directory under a user-defined volume on the HM (user volume names must begin with a character other than ! or &). Each directory can have one or more files assigned to it. A volume can have up to 63 directories. No wild card characters or options are applicable with this request.

4.10.11.1 Create Directory Request

Header information:
Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..8—Reserved for future use
Word 9—Command type = 10
Word 10—Not used
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the user volume directory (NET>vol dir)
Words 27..40—Not used

4.10.11.2 Create Directory Response

Header information:
Word 1—Transaction Code = 63
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus (0 = normal)
Words 13..40—Request echo

4.10.12 Delete Directory

This request/response pair is used to delete a directory from a user-defined volume on the HM (user volume names must begin with a character other than ! or &). A directory that still contains files cannot be deleted. No wild card characters or options are applicable.

4.10.12.1 Delete Directory Request

Header information:
Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..8—Reserved for future use
Word 9—Command type = 11
Word 10—Not used
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the user volume directory (NET>dir)
Words 27..40—Not used

4.10.12.2 Delete Directory Response

Header information:
Word 1—Transaction Code = 63
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus (0 = normal)
Words 13..40—Request echo

4.10.13 Delete Files

This request/response pair is used to delete one or more files from a user-defined volume on the HM (user volume names must begin with a character other than ! or &). **A deleted file cannot be recovered.** Wild card characters can be used to allow selective and multiple file deletions. The -D option lists the files deleted in the Data Out file if specified.

4.10.13.1 Delete Files Request

Header information:
Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..8—Reserved for future use
Word 9—Command type = 12
Word 10—Not used
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the user volume file
(NET>vdir>filename.xx or NET:vdir>*.* -D)
Words 27..40—Not used

4.10.13.2 Delete Files Response

Header information:
Word 1—Transaction Code = 63
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus (0 = normal)
Words 13..40—Request echo

4.10.14 Break Request

This request/response pair is used to abort a previous file utility transaction command. The return status in the response message of the aborted request will contain code value 12 (XFR_VAX_REQUESTED_ABORT).

4.10.14.1 Break Request

Header information:
Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format:
Words 1..8—Reserved for future use
Word 9—Command type = 13
Word 10—Not used
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..40—Not used

4.10.14.2 Break Response

Header information: Word 1—Transaction Code = 63
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format: Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus (0 = normal)
Words 13..40—Request echo

4.10.15 Data Out Request

This request/response pair is used to turn on (assign) or turn off (de-assign) the Data Out file function or to request its on/off status. While the Data Out file is assigned, the file utility transactions that list data will put ASCII text into that file. The "results" text file can be transferred to the host by the Read File request.

The request uses one of three sub-command values depending on the action desired.

- Sub-command value 0 (OFF)—Deassigns the "results" text file.
- Sub-command value 1 (ON)—Specifies (assigns) the pathname of the "results" text file.
- Sub-command value 2 (STATUS)—Requests the on/off status of the "results" text file and, if it is "on," returns the file pathname.

4.10.15.1 Data Out Request

Header information: Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format: Words 1..8—Reserved for future use
Word 9—Command type = 14
Word 10—Sub-command (0 = OFF, 1 = ON, 2 = STATUS)
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname (PN:nn>vdir>filename.xx or NET>vdir>filename.xx) for the user volume file (required by sub-command 1)
Words 27..40—Reserved for response data

4.10.15.2 Data Out Response

Header information: Word 1—Transaction Code = 63
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format: Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus
(0 = Data Out OFF, 1 = Data Out ON)
Words 13..26—Request echo
Words 27..40—Data Out file's pathname
(NET>vdir>filename.yy) returned when the
subcommand value = 1 or 2

4.10.16 Retrieve File Names and Extensions

This request/response pair is used to request the file names and extensions that are associated with an LCN NET/PN volume or directory. A maximum of 1180 file names (with their associated extensions) are returned.

4.10.16.1 Retrieve File Names and Extensions Request

Header information: Word 1—Transaction Code = 62
Word 2—Number of Words = 40
Word 5—CG Identifier = 2

Message format: Words 1..8—Reserved for future use
Word 9—Command type = 15
Word 10—Not used
Word 11—Reserved for response data
Word 12—Reserved for response data
Words 13..26—LCN pathname for the NET or PN
volume/directory file name.
(NET>vdir>*.* or
NET>vdir>filename.* or
NET>vdir>*.NN—where .NN is a specific suffix)
Words 27..40—Not used

4.10.16.2 Retrieve File Names and Extensions Response

Header information:
Word 1—Transaction Code = 63
Word 2—Number of Words = 40 + 5900
Word 5—CG Identifier = 2

Message format:
Words 1..10—Request echo
Word 11—Return status (0 = normal)
Word 12—Return substatus
(0 = normal, 1 = 1180 names limit has been reached and not all names could be returned)
Words 13..26—Request echo
Word 27—Number of file names returned
Words 28..40—Not used

Data information:
Words 1..5900—File names and extensions (1180 maximum)
The data format for each file name and extension consists of five consecutive words as follows:
- Words 1..4—File name (8 ASCII characters)
- Word 5—File extension (2 ASCII characters)

4.10.17 File Transfer Response Status Codes

The following information defines the various major and sub-status response status codes that can be returned by the CG to the various file requests in words 11 and 12 of the response message.

4.10.17.1 Return Status Codes

Following are the return status codes that can be found in message word 11 of the response to a file request:

Code Value	Definition
0	XFR_SUCCESS
1	XFR_FILE_MANAGER_ERROR (See Substatus value for more information)
2	XFR_BUSY
3	XFR_HEADER_ERR
4	XFR_FILE_ALREADY_EXISTS
5	XFR_REC_SIZE_MISMATCH
6	XFR_DATA_DOES_NOT_FIT
7	XFR_DATA_ERROR
8	XFR_INVALID_REC_SIZE
9	XFR_INVALID_REC_NUMS
10	XFR_READ_DATA_ERROR
11	XFR_INVALID_FILE_NAME
12	XFR_VAX_REQUESTED_ABORT
13	XFR_WRITE_OUT_OF_SEQUENCE (Write transmission counter in error)
14	XFR_UTILITY_ERROR (See Substatus value for more information)
15	XFR_NO_DATA_OUT_FILE

4.10.17.2 Return Substatus Codes When the Status Code Equals 1

The following File Manager substatus codes can be found in message word 12 of the response to a file request when Return Status (message word 11) equals 1.

Code Value	Definition
0	$FMS_In Progress
1	$FMS_End_Of_File
2	$FMS_Timeout_Expiration
3	$FMS_Data_Hard_Error
4	$FMS_FAB_Hard_Error
5	$FMS_Dir_Hard_Error
6	$FMS_Inconsistent_Command
7	$FMS_Device_Timeout
8	$FMS_Invalid_Command
9	$FMS_Invalid_LRN
10	$FMS_Open_Files_Exhausted
11	$FMS_LRNs_Exhausted
12	$FMS_LRN_Unassigned
13	$FMS_File_Access_Denied
14	$FMS_Device_Access_Denied
15	$FMS_Reserved_Status_1
16	$FMS_Incompatible_Option
17	$FMS_Invalid_Filename

```
18      $FMS_Duplicate_Filename
19      $FMS_File_Not_Found
20      $FMS_Device_Not_Found
21      $FMS_Access_Violation
22      $FMS_Invalid_Buffer_Length
23      $FMS_Reserved_Status_2
24      $FMS_Reserved_Status_3
25      $FMS_Reserved_Status_4
26      $FMS_Insufficient_Storage_Space
27      $FMS_No_Local_Request_Class_System_Memory
28      $FMS_End_Of_Directory
29      $FMS_Reserved_Status_5
30      $FMS_Reserved_Status_6
31      $FMS_Buffer_Overflow
32      $FMS_Volume_Not_Mounted
33      $FMS_Reserved_Status_8
34      $FMS_File_In_Use
35      $FMS_Volume_In_Use
36      $FMS_Reserved_Status_10
37      $FMS_Reserved_Status_11
38      $FMS_Internal_Error
38      $FMS_Reserved_Status_12
40      $FMS_Reserved_Status_13
41      $FMS_Reserved_Status_14
42      $FMS_Reserved_Status_15
43      $FMS_Size_Conflict
44      $FMS_Invalid_Variable_Record_Length
45      $FMS_Reserved_Status_16
46      $FMS_File_Not_Open
47      $FMS_Reserved_Status_17
48      $FMS_Unspecified_Device_Error
49      $FMS_Invalid_Buffer_Address
50      $FMS_LRN_Not_On_Volume
51      $FMS_Success
52      $FMS_Invalid_Volume_Name
53      $FMS_Volume_Not_Found
54      $FMS_Duplicate_Volume
55      $FMS_Duplicate_Volume_Alias
56      $FMS_Volume_Alias_Not_Found
57      $FMS_Logical_Device_Not_Found
58      $FMS_Duplicate_Logical_Device
59      $FMS_No_Local_Request_Class_User_Memory
60      $FMS_No_Remote_Request_Class_Memory
61      $FMS_Heap_Manager_Failure
62      $FMS_Unimplemented_Function
63      $FMS_Attributes_Incompatible
64      $FMS_LRN_Cancelled
65      $FMS_LRN_Failed
66      $FMS_Illegal_LRN_Deallocation
67      $FMS_LDIDs_Exhausted
68      $FMS_Invalid_Device_ID
69      $FMS_Invalid_CRB_Identifier
70      $FMS_Unsupported_Feature_On_Personality
71      $FMS_Corrupted_Directory_Data
72      $FMS_VVAT_Table_Full
73      $FMS_Maximum_Tracks_Exceeded
74      $FMS_LRN_Allocation_Denied
```

75	$FMS_Device_Failed
76	$FMS_Request_Class_Table_Error
77	$FMS_Invalid_Request_Class_Data
78	$FMS_Volume_Access_Denied
79	$FMS_Invalid_Physical_Node_Identifier
80	$FMS_Invalid_File_Configuration_Revision
81	$FMS_Volume_Alias_Not_Empty
82	$FMS_Device_Not_Redundant
83	$FMS_Redundant_Device_Not_Available
84	$FMS_Illegal_Device_State_Transition
85	$FMS_Option_Not_Purchased
86	$FMS_Invalid_Device_Address
87	$FMS_Descriptors_Not_Found
88	$FMS_Remote_LCN_Has_Not_Connected_to_Local_LCN
89	$FMS_Local_LCN_Has_Not_Connected_to_Remote_LCN
90	$FMS_Volume_Read_Permission_on_Remote_LCN_Denied
91	$FMS_Volume_Read_Write_Permission_on_Remote_LCN_Denied
92	$FMS_Specified_Remote_LCN_Not_Defined
93	$FMS_Not_Mutually_Connected_to_Remote_LCN

4.10.17.3 Return Substatus Codes When the Status Code Equals 14

The following Utility Error substatus codes can be found in message word 12 of the response to a file request when Return Status (message word 11) equals 14.

Code Value	Definition
3	U$Incomplete_Command_Msg
4	U$Invalid_Pathname_Msg
6	U$No_Files_On_Volume_Msg
8	U$Invalid_Option_Msg
9	U$Invalid_Command_Format_Msg
21	U$Extra_Chars_Msg
22	U$Out_Of_Memory_Msg
24	U$Function_not_implemented_msg
30	U$Illegal_Use_Of_Wildcard_Msg
32	U$Bad_Destination_File_Size_Msg
33	U$Destination_Pathname_Required_Msg
34	U$Source_File_Error_Msg
35	U$Destination_File_Error_Msg
36	U$Temporary_File_Error_Msg
37	U$Max_Files_Out_Of_Range_Msg
38	U$Invalid_Directory_Msg
39	U$Blocksize_Out_Of_Range_Msg
40	U$Destination_File_Ext_Not_Allowed
53	U$Illegal_Pathform_For_Create_Msg
56	U$File_Manager_Pointer_Error_Msg
57	U$Can_Not_Rename_Volume_on_HM_Msg
63	U$Bad_Source_Drive_Number
67	U$Bad_Destination_Drive_Number
68	U$Destination_Same_As_Source
70	U$Error_In_Perform_PIO
73	U$Memory_Allocation_Error
77	U$Invalid_Physical_Node_Number_Msg
81	U$Invalid_Drive_Number_Msg
82	U$Synchronization_Was_Initiated_Msg
91	U$Duplicate_Volume_ID_Msg

92	U$FMD_Too_Long
94	U$Dismount_Service_Err_Msg
95	U$Not_A_Local_Device
97	U$Physical_Node_Nonexistant
98	U$Cannot_Find_HM_Nodes_On_Network
99	U$No_Running_HM_Nodes_On_Network
101	U$Max_Memory_Directory_Files_Msg
102	U$Mem_Directory_no_BS_Needed_Msg
103	U$Device_Not_Present
104	U$Device_Failed
105	U$Device_Offline
106	U$Volume_Corrupted
107	U$Device_Not_Formatted
108	U$Device_Not_Mounted
109	U$Device_Resource_Error
110	U$Device_Syncronizing
111	U$Device_Degraded
112	U$Device_Formatting
113	U$Device_State_Unknown

DATA LINK CHARACTERISTICS
Section 5

This section summarizes the data link hardware, transaction management, and supported protocols (Bisynch and HDLC). See Appendix B for cabling and connector information.

5.1 INTRODUCTION

Communication between the host processor and the CG is over a serial link, using either Binary Synchronous (Bisynch) or HDLC protocol. When Bisynch is used, either one or two half-duplex links can be used (one is recommended). When HDLC is used, one full-duplex link is supported. You are expected to provide a compatible data link package for your host processor.

5.2 HARDWARE INTERFACES

The following hardware interfaces are supported:

RS232C — EIA Standard RS232C connectors are used. Cable lengths greater than 50 feet require a modem. Speeds are: 1200, 1760, 2152, 2400, 4800, 9600, or 19200 baud.

RS422 — RS449 37-pin connectors are used. For cable lengths up to 4000 feet. Full duplex channels are supported. Speeds are 1200, 1760, 2152, 2400, 4800, 9600, 19200, 38400, 57600 (HDLC limit), and 76800 baud.

Circuits used:

- Shield (connected to the CG end only)
- Signal Ground (circuit SG)
- Send Data (circuit SD)
- Receive Data (circuit RD)
- Data Mode (circuit DM)
- Terminal Ready (circuit TR)

5.3 BISYNCH PROTOCOL

Binary Synchronous Communication Protocol Procedure for Point-to-Point Operation with Contention, as described in the IBM publication GA27-3004-2, *General Information—Binary Synchronous Communications*, is provided.

This implementation uses transparent Bisynch with EBCDIC-control characters (see Table 5-1 for hexadecimal equivalents for the EBCDIC-control characters). The maximum transmission block size is 512 words (1024 bytes). Both single- and multiple-block transactions are supported (13 block maximum). The Bisynch header is not used, instead a message-specific header is included as part of the ASCII character-text message in each block. See Section 4 for details of the individual message formats.

The following message formats are supported:

- Line Bid Sequence
- Single Block Text Transfer
- Multiple Block Text Transfer
- Wait Before Transmitting Positive Acknowledge
- Start Transparent Text (DLE STX)
- End Transparent Text (DLE ETX)
- End Transparent Text Block (DLE ETB)
- Synch/Fill (DLE SYN)
- Ignore DLE Pattern (DLE DLE)

CAUTION

Be sure that your CG is configured to correspond with the host data-link type (single/dual).

Table 5-1 — Hexadecimal Equivalents for EBCIDIC Control Characters

CONTROL CHARACTER	HEXADECIMAL CODE	CONTROL CHARACTER	HEXADECIMAL CODE	CONTROL CHARACTER	HEXADECIMAL CODE
SYN	32	STX	02	ETB	26
ETX	03	EOT	37	ENQ	2D
ACK0	10 70	ACK1	10 61	NAK	3D
DLE	10	WACK	10 6B	RVI	10 7C
TTD	02 2D				

NOTE

The “most significant” byte (MSB) of each data word (16 bits) is transmitted first.

5.3.1 Single-Link Bisynch Sequences

Most CG applications are expected to use single-link communications. Each single-link transaction starts with the line-bid sequence and ends with release of the line. The CG conflict timeout has been set to 0.75 second to reduce delays. If possible, the host computer timeout should be set similarly (for CM60 applications, the DPS6 timeout is set at one second). The CG is assigned primary priority for contention resolution.

Simplified example of single-block text transfer for single link:

Line Bid Sequence

```
SENDER      ENQ
RECEIVER          ACK0
```

Single Block Text Transfer

```
SENDER      DLE STX "transparent text" DLE ETX BCC
RECEIVER                                           ACK1
```

End Transmission

```
SENDER      EOT
```

Example of multiblock text transfer for single link:

Line Bid Sequence

```
SENDER      ENQ
RECEIVER          ACK0
```

First Block Text Transfer

```
SENDER      DLE STX "transparent text" DLE ETB BCC
RECEIVER                                           ACK1
```

Intermediate Blocks Text Transfer

```
SENDER      DLE STX "transparent text" DLE ETB BCC
RECEIVER                                           ACK*
```

Last Block Text Transfer

```
SENDER      DLE STX "transparent text" DLE ETX BCC
RECEIVER                                           ACK*
```

End Transmission

```
SENDER      EOT
```

NOTES: 1) ACK* means the correct ACK0 or ACK1, which alternate.
2) NAK received in place of ACK* requires retransmission of that block.

5.3.2 Dual-Link Bisynch Sequences

Although a single-link connection to the host computer is recommended, there is an option for a dual Bisynch-link connection. The advantage of dual links is improved throughput rates. The disadvantage is the need for more complex host-computer software necessary to react to the CG on startup and in failure recovery conditions and the maintenance of send and receive maintenance counters.

In a dual-link configuration, the CG checks the Message Counter of each message. It must be between 1 and 30,000 and be different from the last transmission of any data message (duplicates are ignored). The host processor should perform the same checks.

For the dual-link configuration, the line-bid sequence occurs only at startup. The lines are then kept open using TTD (Temporary Text Delay). Line two is used for host-processor transmissions and line one for CG transmissions. Note that these are half-duplex transmissions with the protocol messages (ACK, NAK) returned over the same line they were received on. The resulting throughput is close to that of a full-duplex implementation because there are no line bidding delays.

Examples of Dual Link sequences:

Line Bid (at startup only)

```
SENDER      ENQ
RECEIVER          ACK0
```

Line Idle Mode

```
SENDER      TTD
RECEIVER          NAK
```

Message Transfer (single block)

```
SENDER      DLE STX "transparent text" DLE ETX BCC
RECEIVER                                          ACK*
```

Multiblock Message Transfer (first and intermediate blocks)

```
SENDER      DLE STX "transparent text" DLE ETB BCC
RECEIVER                                          ACK*
```

Multiblock Message Transfer (final block)

```
SENDER      DLE STX "transparent text" DLE ETX BCC
RECEIVER                                          ACK*
```

NOTES:
1) ACK* means the correct ACK0 or ACK1, which alternate.
2) Idle Mode exchanges (TTD-NAK) can take place between blocks of a multi-block transfer.
3) NAK received in place of ACK* requires retransmission of that block.

5.4 HDLC LINK PROTOCOL

Operation of the HDLC link follows the Link Access Procedure, Balanced (LAPB), according to C.C.I.T.T. document AP VIII-58-E June 1984 (Recommendation X.25) and complies with FIPS 100 certification.

To provide a compatible HDLC interface with the CG, host processor software must meet the following requirements:

- Link protocol is HDLC LAPB.
- Station identifier is configurable as either 1 or 3; default is 1.
- Maximum message size is 1024 bytes. Maximum frame size is 1030 bytes, consisting of: Start Flag (1 byte), Frame Header (2 bytes), Message (1024 bytes), CRC (2 bytes), and End Flag (1 byte).
- Window size (maximum number of outstanding frames) is set to 1.
- Time to wait (T1) for acknowledgment to a command frame varies from 0.1 second to 25.5 seconds with a default of 1.6 seconds. This time must be long enough to exchange a complete frame.
- Maximum number of transmissions of a command frame following the run-out of timer T1 ranges from 2 to 50 with a default of 10. After the specified number of retries with no response, a disconnect occurs.

The communication interface software must support interface routines to

- communicate with driver routines
- control DTR and detect DSR state

5.5 MESSAGE STRUCTURE

The message structure is similar for both Bisynch and HDLC protocols. Each message is packeted into 512-word (1024-byte) blocks.

Each block begins with an 8-word header. The remainder of the block contains message information words up to the maximum block size. The maximum number of blocks in a message is 13 blocks. This provides for a maximum of 6552 information (nonheader) words for each message.

The 8-word header with each block contains the following information:

Word 1—Transaction Code (integer)
Word 2—Number of Words (integer, 1..504)
Word 3—Number of Blocks (integer, 1..13)
Word 4—Block Number (integer, 1..13
Word 5—CG Identifier (integer)
Word 6—Host Task Identifier (2 ASCII characters)
Word 7—Message Tag (integer)
Word 8—Message Counter (integer, 1..30000)

Header words 1, 5, and 6 are assigned at the application level and are not modified by the communications package. See paragraph 4.1.2.1 for a description of header contents.

5.6 HOST PROCESSOR STATE DIAGRAM

The types of messages that should be expected from, or are accepted by, the CG vary as processing states of the host-processor change. Figure 5-1 summarizes the expected reaction to various CG messages depending upon host-processor processing state. Those processing states are defined as follows:

INIT	Hardware-level communication has been established; waiting for Restore Message from CG. Will also receive Time Synch messages in this and all other states.
RESTORE 0	Startup processing; ready to transmit the next message in the sequence.
RESTORE 1	Startup processing; waiting for Confirmation message from CG following a List Request or Restore Complete message.
NORM 0	Normal processing; data messages can be sent/received.
NORM 1	Normal processing; waiting for Confirmation message from CG following a Data Message. Data messages can be received.

Figure 5-1 — Host Processor Processing States

	STARTUP			NORMAL OPERATION	
EVENT or MESSAGE \ STATE	INIT	Restore 0	RESTORE 1	NORMAL 0	NORMAL 1
Time Synch Message (31) Received	Return Time Synch Message to CIU	Return Time Synch Message to CIU	Return Time Synch Message to CIU	Return Time Synch Message to CIU	Return Time Synch Message to CIU
Restore Message (43) Received	Send CNFPOS and goto RESTORE 0 OR Send CNFNEG	Not appropriate (Note 1)	Not appropriate (Note 1)	Not appropriate (Note 1)	Not appropriate (Note 1)
Data Message Received	Not appropriate (NOTE 2)	Send CNFPOS OR Send CNFNEG	Send CNFPOS OR Send CNFNEG	Send CNFPOS OR Send CNFNEG	Send CNFPOS OR Send CNFNEG
Positive Confirm (30) Received	Not appropriate (NOTE 3)	Not appropriate (Note 3)	goto RESTORE 0 OR (if Restore is complete) goto NORMAL 0	Not appropriate (Note 3)	goto normal 0
Negative Confirm (30) Received	Not appropriate (NOTE 4)	Not appropriate (Note 4)	Retransmit previous message (NOTE 7)	Not appropriate (Note 4)	Retransmit previous message (NOTE 7)
Data messages to CIU stacked at Host Processor	Not appropriate (NOTE 5a)	Send next data message (Note 5b) goto RESTORE 1	No action (Note 5c)	Send next data message and goto NORMAL 1	No action (Note 5c)
Link Fail (32) Message Received (Note 6)	goto INIT 0	goto INIT 0	goto INIT 0	goto INIT 0	goto INIT 0

3231

5.6.1 Explanatory Notes to Figure 5-1

Note 1 RESTORE message is expected only while host processor is in INIT state. Receipt of this message, at any other time, indicates violation of the 3-minute time limit between dispatch of a RESTORE and receipt of a RESTORE COMPLETE, by the CG.

Note 2 Data messages occur only in response to requests from the host processor and cannot occur in this state.

Note 3 Each Data Message ultimately must be responded to with a Positive Confirmation Message (this excludes only Time Synch and the Confirmation messages themselves). No additional messages can be sent until the confirmation is received.

Note 4 Negative Confirmation messages can occur only following the sending of a Data Message.

Note 5a No Data Messages can be sent from this state.

5b The Data Messages sent from this state are those required for database initialization and end with a Restore Complete message. Any message types are confirmed by the CG, but ignored.

5c No message can be sent until the previous message has received a Positive Confirmation from the CG.

Note 6 Applies to only Dual Bisynch Link configuration. Host processor should alarm link failure and wait for hardware-level communication to be resumed on both links.

Note 7 The Message Counter must be incremented before retransmission of a message over a dual link.

5.6.2 Analysis of Data Link Messages

Figure 5-2 is a re-creation of a series of single-link messages between a CG and a host processor recorded by a data analyzer at the beginning of a startup sequence. The transmissions are shown in the order they occurred, with the host-processor transmissions shown underlined.

To aid in your understanding of the messages, see Table 5-1 for Hex code equivalents for the Bisynch-control characters and Table 5-2 for Hex code equivalents of the Message Transaction Codes.

The messages are discussed in their order of occurrence.

1) ENQ from the CG to bid for the line. Note the leading (SYN) and trailing (FF) pad characters. Five leading and four trailing pad characters are recommended. Fewer may work depending on link speed and other variables.

2) ACK0 response from the host processor.

3) Time Synchronization Message from the CG (most of data field not shown). Message is composed of:

   - Six leading pad characters
   - DLE STX to mark start of transparent text
   - Eight header words lead by Time Synch Transaction Code
   - DLE ETX to mark end of transparent text
   - Block Check Character (two bytes)
   - Trailing pad character

4) ACK1 response from the host processor.

5) EOT from the CG to end transmission.

6) ENQ from the host to bid for the line.

7) ACK0 response from the CG.

8) Echo back of the Time Synchronization Message by the host.

9) ACK1 response from the CG.

10) EOT from the host to end transmission.

11) ENQ from the CG to bid for the line.

12) ACK0 response from the host.

13) CG Restore Message (Transmission Code 43 decimal) from the CG.

14) ACK1 response from the host.

15) EOT from the CG to end transmission.

16) ENQ from the host to bid for the line.

17) ACK0 response from the CG.

18) Confirmation Message (Transmission Code 30 decimal) from the host to indicate successful receipt of the Restore Message.

19) ACK1 response from the CG.

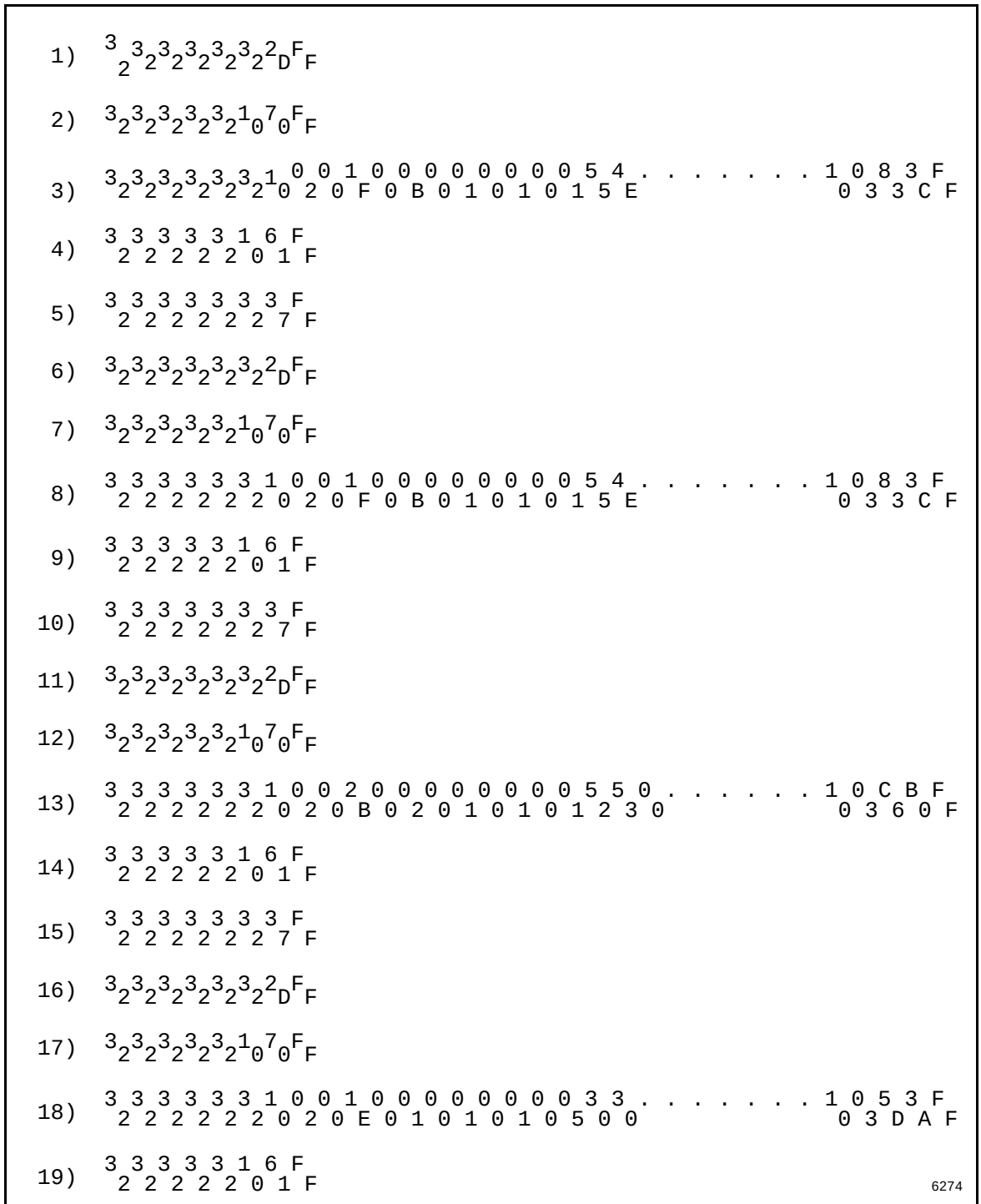


Figure 5-2 — Data Analyzer Output

Table 5-2 — Hexadecimal Values for Message Transaction Codes

TRANSACTION CODE VALUES HEX	DECIMAL	MESSAGE NAME	WHERE DESCRIBED	MESSAGE INITIATOR
1	1	Operator Message Confirmation/Timeout	4.9.5	CG
2	2	Turn On ACP	4.4.5	CG
3	3	Get Data Request (internal DDT in CG)	4.6.1	host
4	4	Get Data Request (internal DDT in host)	4.6.2	host
5	5	Store Data Request	4.6.4	host
6	6	Build internal DDT Request	4.5.1	host
7	7	Get Data Return	4.6.3	CG
8	8	Store Data Return	4.6.5	CG
9	9	Build internal DDT Return	4.5.2	CG
A	10	Store internal DDT in CG Request	4.5.3	host
B	11	Delete internal DDT from CG Request	4.5.4	host
C	12	Internal DDT Store/Delete Return	4.5.5	CG
D	13	Get Message Request	4.9.1	host
E	14	Get Message Return	4.9.2	CG
F	15	Send Message Request	4.9.3	host
10	16	Send Message Return-Immediate	4.9.4	CG
15	21	ACIDP Execution State Change Request	4.4.6	host
16	22	ACIDP Program Mode Change Request	4.4.1	host
17	23	ACIDP Mode Change Return	4.4.2	CG
18	24	Get ACIDP Status Request	4.4.3	host
19	25	Get ACIDP Status Return	4.4.4	CG
1C	28	Get Single Value Request (ext addr)	4.7.2	host
1D	29	Get Single Value Return (ext addr)	4.7.4	CG
1E	30	Message Confirmation	4.2.2	either
1F	31	Time Synchronization	4.2.1	either*
20	32	Data Link Failure	4.2.3	either
23	35	Store Single Value Request (ext addr)	4.7.5	host
24	36	Store Single Value Return (ext addr)	4.7.6	CG
25	37	Store Single Value Request (int addr)	4.7.11	host
26	38	Store Single Value Return (int addr)	4.7.12	CG
27	39	Get Single Value Request (int addr)	4.7.9	host
28	40	Get Single Value Return (int addr)	4.7.10	CG
29	41	Convert Identifier Request	4.7.7	host
2A	42	Convert Identifier Return	4.7.8	CG
2B	43	CG Restore	4.3.1	CG
2C	44	Restore Complete	4.3.2	host
2D	45	Internal DDT Connect/Disconnect Request	4.4.7	host
2E	46	Internal DDT Connect/Disconnect Return	4.4.8	CG
2F	47	CG Database List Request	4.3.3	host
30	48	ACIDP/CRDP/DDT Tables List Return	4.3.4-.8	CG
32	50	Get History Request	4.8.1	host
33	51	Get History Return	4.8.2	CG
3E	62	File Transfer Request	4.10	host
3F	63	File Transfer Response	4.10	CG

*Time Synchronization message from host always is echo back of CG message.

(Continued)

Table 5-2 — Hexadecimal Values for Message Transaction Codes (continued**)**

TRANSACTION CODE VALUES HEX	DECIMAL	MESSAGE NAME	WHERE DESCRIBED	MESSAGE INITIATOR
65	101	Operator Message Confirmation/Timeout	4.9.5	CG
66	102	Turn On ACP	4.4.5	CG
67	103	Get Data Request (internal DDT in CG)	4.6.1	host
68	104	Get Data Request (internal DDT in host)	4.6.2	host
69	105	Store Data Request	4.6.4	host
6A	106	Build internal DDT Request	4.5.1	host
6B	107	Get Data Return	4.6.3	CG
6C	108	Store Data Return	4.6.5	CG
71	113	Get Message Request	4.9.1	host
73	115	Send Message Request	4.9.3	host
79	121	ACIDP Execution State Change Request	4.4.6	host
7A	122	ACIDP Program Mode Change Request	4.4.1	host
7C	124	Get ACIDP Status Request	4.4.3	host
7D	125	Get ACIDP Status Return	4.4.4	CG
80	128	Get Single Value Request (ext addr)	4.7.2	host
81	129	Get Single Value Return (ext addr)	4.7.4	CG
87	135	Store Single Value Request (ext addr)	4.7.5	host
88	136	Store Single Value Return (ext addr)	4.7.6	CG
89	137	Store Single Value Request (int addr)	4.7.11	host
8A	138	Store Single Value Return (int addr)	4.7.12	CG
8B	139	Get Single Value Request (int addr)	4.7.9	host
8C	140	Get Single Value Return (int addr)	4.7.10	CG
8D	141	Convert Identifier Request	4.7.7	host
8E	142	Convert Identifier Return	4.7.8	CG
91	145	Internal DDT Connect/Disconnect Request	4.4.7	host
92	146	Internal DDT Connect/Disconnect Return	4.4.8	CG
93	147	CG Database List Request	4.3.3	host
94	148	ACIDP/CRDP/DDT Tables List Return	4.3.4-.8	CG
96	150	Get History Request	4.8.1	host
97	151	Get History Return	4.8.2	CG

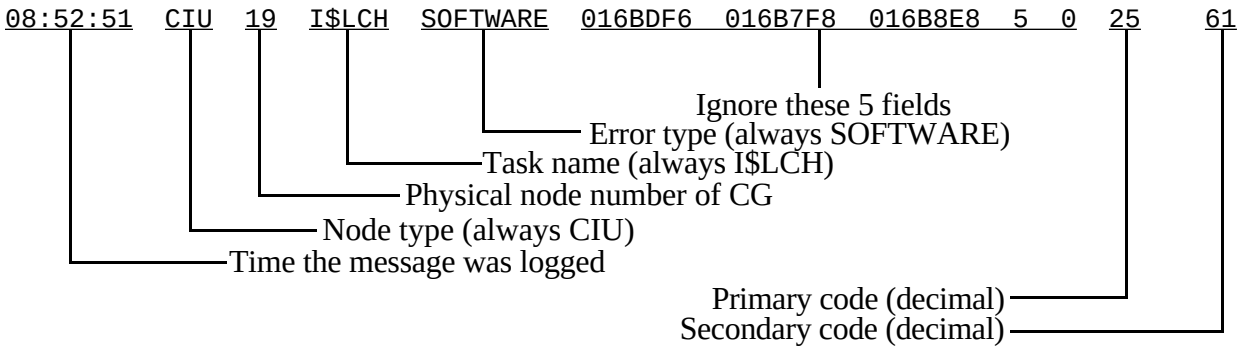
5.7 SYSTEM ERROR JOURNAL REPORTS

Reports are issued by the CG to the LCN System Error Journal in the event of certain data link communications errors. Each error type is assigned both a primary and secondary code for identification as follows.

Primary code	Secondary code	Meaning
0	79	Both links failed
0	80	Link 1 up
0	81	Link 2 up
0	82	Both links up
25	61	Block out of sequence
26	62	Illegal Transaction Code
27	63	Illegal CG Identifier
28	64	Number of words out of range
29	65	Number of blocks out of range
30	66	Block data too small
31	67	Block data size conflict
32	68	Block header conflict
33	69	Message size conflict
34	70	Message timeout
35	71	Not multimessage type
37	73	Confirm timeout
38	74	Time Synchronization timeout
39	75	Entity not found
40	76	Invalid Internal ID

Note that the meanings assigned to codes 25/61 (primary/secondary) through 34/70 correspond to negative Message Confirmation codes 0 through 9 (see paragraph 4.2.2).

An example showing the format of these messages follows.



CG DATABASE CONTENT
Section 6

This section covers the content and use of information in the CG database.

6.1 INTRODUCTION

The three user-specified components of the CG database are Advanced Control Interface Data Points (ACIDP) with their associated Custom Data Segments, the Calculated Results Data Points (CRDP), and the Internal Data Tables (internal DDT).

Creation of the Internal Data Tables is explained in paragraph 4.5, while ACIDP and CRDP preparation are explained in Appendix E. An additional user specification that affects the CG database is the assignment of Process Units, explained in Appendix D.

6.2 CG MEMORY ESTIMATING

There are two sets of limits that must be considered in preparing the CG database. First is the set of design limits on the numbers of each items that are allowed. Second is the amount of memory available and the size requirements for each item.

The built-in design limits (further restricted by memory availability) are

250 ACIDPs
500 CRDPs
10 Custom Data Segments per ACIDP or CRDP
40 CG-resident internal DDTs
300 parameters in a internal DDT
63 Units assigned to a CG

6.2.1 Memory Management Schematic

The amount of memory available in the CG for database use varies by product release. You can use the schematic display CGDBSIZE to determine if the present use of memory in a CG is such that you may need to install additional memory. This display is provided in directory DIA1 on cartridge &CR6 or on floppy diskette DIA1 in Media Kit MP-MKUS10 or MP-MKUS20.

CGDBSIZE is available at Universal Stations running the Operator Personality when the pathname for the volume or directory that contains this schematic is listed as one of the schematic pathnames in your Pathname Catalog in the Area Database. When this is true, you can press the SCHEM key, type in the name of the schematic, CGDBSIZE, and press ENTER. The schematic contains instructions for its use when displayed.

You can copy the schematic to an existing volume that is already listed in a Pathname Catalog, or you can add its pathname to a Pathname Catalog and then do an Area Change at a US to make it available. Detailed instructions for adding a schematic to a Pathname Catalog are under subsection 29.2 in the *Engineer's Reference Manual*, in the *Implementation/Startup and Reconfiguration - 2* binder.

6.2.2 Database Memory Requirements

Database memory requirements are

ACIDP—143 words each (plus CDS requirements)
CRDP—36 words each
internal DDT—40 + (25*the number of items in table) words each
Custom Data Segments —three factors are involved:

- Control Structure: 80 words of control structure are added to each ACIDP or CRDP with one or more CDS.
- Descriptor Segment: 13 + (22 times the number of parameters) words are required by each CDS for its descriptor segment.
- Data space: CDS data use varies with the mix of parameter types.
  Each Real parameter requires 2 words
  Each ASCII parameter requires 21 words
  Each Enumeration parameter requires 2 words
  Each Self-defining enumeration requires 7 words
  Each Boolean parameter requires 1 word

Process Units—300 words for each 150 points (or fraction of 150) in the same Unit.

6.3 VIEWING AND CHANGING ACIDP AND CRDP POINTS

6.3.1 Viewing and Changing Parameter Values From the Universal Station

The current values of certain ACIDP and CRDP parameters can be viewed—and some scheduling-related parameters can be changed—from the area Universal Station in the Operator Personality, through the Group and Detail displays of these points. You can also create custom Universal Station displays with similar capabilities (see the *Picture Editor Reference Manual* for details).

Universal Station display and change access to ACIDP parameters is summarized in Table 6-1. US display access to CRDP parameters is summarized in Table 6-2. Brief descriptions of these parameters—sorted by parameter name—follow at paragraph 6.4. A more complete description of all CG parameters is found in the *Computer Gateway Parameter Reference Dictionary.*

6.3.2 Viewing and Changing Parameter Values From the Host Processor

ACIDP parameter values can be obtained by the host processor through use of the "Get ACIDP Status" request (paragraph 4.4.3). There is no equivalent access to the CRDP parameters.

There is no mechanism for CRDP parameter change by the host processor, and ACIDP parameter change is limited to the following:

EXECSTAT	can be changed by using the "ACIDP Execution State Change" request message (paragraph 4.4.6).
PROGSTAT	can be changed by using the "ACIDP Program Mode Change" request message (paragraph 4.4.1).
PPS	can be changed by using a "Store Single Value" request message addressed to the ACIDP (name.PPS) to store the enumeration ON (this is used by one ACP to force another's execution).
TAKE_IP	can be changed by using a "Store Single Value" request message addressed to the ACIDP (name.TAKE_IP) to store the enumeration ON (this is used by one ACP to tell another to take its initialization path the next time it is turned on).

Table 6-1 — Universal Station Display and Change of ACIDP Parameters

PARAMETER NAME	DETAIL DISPLAY		GROUP DISPLAY	
	DISPLAY	CHANGE	DISPLAY	CHANGE
ABORTCOD	no*	no	no	no
ACCESKEY	yes	no	no	no
ACPROG	yes	no	no	no
ACT_TYPE	yes	no	no	no
CONF_RQD	no	no	no	no
CONFWAIT	yes	no	no	no
EXECSTAT	yes	no	yes	no
INH_STAT	yes	operator	no	no
KEYWORD	yes	no	yes	no
NAME	yes	no	yes	no
NEXT_RTM	yes	no	no	no
OPER_DMD	yes**	operator	no	no
PROGSTAT	yes	no	no	no
PTDESC	yes	no	yes***	no
RTPERIOD	yes	no	no	no
RUN_INIT	yes	no	no	no
STIME	yes	no	no	no
TAKE_I_P	no	no	no	no
UNIT	no	no	no	no

* When nonzero, visible as EXECSTAT value
** Target used to request ACP activation
*** Only shown when point is selected

Table 6-2 — Universal Station Display of CRDP Parameters

PARAMETER NAME	DETAIL DISPLAY		GROUP DISPLAY	
	DISPLAY	CHANGE	DISPLAY	CHANGE
KEYWORD	yes	no	yes	no
NAME	yes	no	yes	no
PTDESC	yes	no	no	no
UNIT	yes	no	no	no

6.4 ACIDP/CRDP PARAMETER DESCRIPTIONS

The following are brief descriptions of the ACIDP and CRDP parameters that are visible through a Universal Station's standard displays. For details of all CG parameters, see the *Computer Gateway Parameter Reference Dictionary.*

ABORTCOD—Four ASCII characters indicating the reason for termination of ACP execution by the host processor. The value 0000 (zeros) is used to designate a normal ACP termination. Other value assignments are defined by the host processor. See EXECSTAT for display information.

ACCESSKEY—Determines whether or not the ACP can execute writes to the LCN. Values are READWRIT or READONLY.

ACPROG—The file name of the ACP as identified within the host processor.

ACT_TYPE—The activation method for the associated ACP. Values are CYCLIC, PERIODIC, CYC_DMD, PER_DMD, or DEMAND.

CONF_RQD—Set ON when an operator message requiring confirmation is sent. Must be true before a message confirmation is processed.

CONFWAIT—Time (in seconds) remaining before a pending message confirmation will time out.

EXECSTAT—The present execution state of the ACP. Values are ABORT, ACCESS, DELAY, OFF, RUN, WAIT, FAIL. When the execution state is ABORT, the ACIDP's Detail Display shows the ABORTCOD value instead.

INH_STAT—An operator-changeable parameter that blocks CG Turn On of the attached ACP. Values are INHIBIT or PERMIT.

KEYWORD—Name shown on Universal Station displays.

NAME—Name of the ACIDP/CRDP.

NEXT_RTM—Next runtime, used by both Periodic and Cyclic activation. The parameter is in form HH:MM:SSbMM:DD:YYb (where "b" indicates a space). Blank if activation type is DEMAND only.

OPER_DMD—An operator-accessible parameter, which when set ON turns on the ACP, but only if the ACT_TYPE value for the ACDIP is either CYC_DMD, PER_DMD, or DEMAND, and ABORTCOD value is zero (see the *Computer Gateway Parameter Reference Dictionary* for effect of OPER_DMD on an Aborted ACP).

PPS—When ON, results in activation of the attached ACP. It can be set ON by an event from the HG or AM, or by a CL/AM program, or an ACP.

PROGSTAT—Installation status of the ACP. Values are NOT_INST, TEST, RESTRICT, NORMAL.

PTDESC—Description of the variable.

RCASENB—Determines whether or not the ACP can execute continuous control by executing SP/OP stores to the LCN. Values are ON and OFF.

RTPERIOD—The time period between runs of a scheduled ACP, in format HH:MM:SS. Minimum period is 10 seconds; maximum period is 24 hours. Not used if ACT_TYPE is DEMAND only.

RUN_INIT—When ON, tells the CG scheduler to turn on the ACP immediately after completing an initialization event (see note on this page).

STIME—The first time of day that a periodic program runs, in format HH:MM:SS (eight ASCII characters). The maximum time is 24:00:00. Not used if ACT_TYPE is CYCLIC, CYC_DMD, or DEMAND.

TAKE_I_P—When ON, informs the attached ACP to take its initialize path. Set ON at any initialization event (see note on this page). Also can be set ON by CL programs and ACPs.

UNIT—Unit identification number.

NOTE

The three types of initialization event are: CG power up and software load (cold restart); host processor initialization or data-link restart (warm restart); and individual ACIDP initialization (connecting it to an ACP or removing abort condition by operator action).

6.5 DATA LINK STATUS INFORMATION

The CG holds data link status information in its Processor Status data point. To view this data, place the point $PRSTSnn (where nn is the CG's node number) in a custom schematic display. The three data link status parameters of this point and their value meanings are:

ULP_STS = IN_SERV—Communications with the host computer have been established. This value is set when Restart is complete.
= FAILED—Communications with the host computer are broken. This value is set when both links have failed.

DL1_STS = IN_SERV—Link 1 has been connected and the CG is using or trying to use the link.
= FAILED—The CG has disconnected Link 1 because of problems.

DL2_STS = IN_SERV—Link 1 has been connected and the CG is using or trying to use the link.
= FAILED—The CG has disconnected Link 2 because of problems.
= NOT_INST—The CG has been configured for operation on Link 1 only.

CG POINT PREPARATION
Appendix A

This appendix summarizes the requirements for preparation of specialized CG data points that regulate the execution of control programs and hold results of control calculations for exchange with other LCN nodes.

A.1 CG POINT BUILDING OVERVIEW

As explained in Section 2, there are two types of data points that reside in the CG, the Advanced Control Interface Data Point (ACIDP) and the Calculated Results Data Point (CRDP). Both point types can be used to store calculated results that are to be exchanged between the host processor and other LCN nodes. The ACIDP also contains the parameters that control the scheduling and execution of an associated ACP in the host processor.

Both point types are built at an Operator Station that is running in the Engineering Personality; however, each point that includes data storage must reference a previously prepared Custom Data Segment (CDS) that defines the special parameters to be added to that point.

A.2 CUSTOM DATA SEGMENT CONSTRUCTION

NOTE

The following information on preparation of Custom Data Segments is intended only as an introduction. Please consult the *Control Language/AM Reference Manual* for details of CL program preparation and the *System Control Functions* manual for additional information on Custom Data Segments.

Custom Data Segments allow you to define new (nonstandard) parameters and add them to data points. Once you define new parameters and add them to data points, they can be accessed in the same manner as standard parameters. Up-to-10 Custom Data Segments can be associated with any ACIDP or CRDP.

Custom Data Segments are constructed as Control Language (CL) Packages, each consisting of a single CDS. These CL "packages" are compiled, then stored on the History Module or on Floppy Disk for use when the individual data points are built. The Data Entity Builder (DEB) is used to add instances of a CDS to one or more data points.

Each CDS consists of a Heading plus one or more parameters.

A.2.1 Custom Data Segment Heading

The Custom Data Segment Heading consists of the word CUSTOM followed by three optional attribute assignments, which change the default values for Class, Access, and Build Visible for this CDS. Either the standard or heading-specified default values are overridden by any individual parameter attribute assignments. Always use the default value for Class when preparing a CDS to be used with an ACIDP or CRDP.

A.2.2 Custom Data Segment Parameters

Each CDS parameter has a heading that begins with the word PARAMETER followed by an up-to-8 character name, an optional data type specifier, and an optional character string to be displayed by the DEB. This is followed by a set of optional attribute assignments.

Data Type can be Number, Time, Logical, Enumeration, String, or Data Point Identifier (or single-dimension arrays of any of these). The default data type is Number.

The parameter attributes are

ACCESS—The Access Attribute defines write access restrictions for the parameter. Read access is never restricted. The access levels are View Only, Operator, Supervisor, Engineer, Program, and Entity Builder. The standard default access level is Engineer. For additional information on parameter access-level significance, see the *System Control Functions* manual.

BLD_VISIBLE—This determines whether or not a preset parameter value can be changed at point-build time. The standard default value is Build Visible.

VALUE—The data type of the constant expression must match the parameter's assigned (or default) type. If no Value is specified and Not Bld_Visible is specified, a default value is assigned. The default values vary by data type as specified in the *Control Language/AM Reference Manual.*

EU—The Engineering Units attribute is a character string that is displayed with other point-parameter information. The default is blanks.

CLASS—For an ACIDP or CRDP CDS, always use the standard default value of General.

A.2.3 Custom Data Segment Example

```
CUSTOM

PARAMETER swdbd1: NUMBER "switch deadband value"
TYPE number
ACCESS engineer
EU "psi"
VALUE 0.5
BLD_VISIBLE

PARAMETER swdbd2
EU "psi"
VALUE 0.5

END CUSTOM
```

Notice that the two parameters generated by this example will be identical in all but their names. There is no name associated with the CDS because it is identified by the name of the file into which it is compiled (only one CDS for each file).

A.2.4 Custom Data Segment Compilation Recommendation

The CL compiler maintains a library file that includes the names of all nonstandard parameter names used in every CDS ever compiled in your system. This file allows for 1000 names, which is normally more than adequate because any of these parameters can be arrays of values and a particular name is entered only once, no matter how many times it is used by multiple Custom Data Segments.

Once a name is entered into the library file, however, there is no way to delete the name. Because it is not desirable to clutter the file with parameter names that were accidentally mistyped, the compiler does not update the library file unless the compiler directive -UL (Update Library) is invoked. You should obey the following sequence when compiling a CDS:

1. First, compile **without** the -UL directive to ensure that the CDS parameters are free of errors. Every new parameter is followed by an error indicating that the -UL option should be used. If any **other** errors appear, they should be corrected.

2. Recompile **with** the -UL directive to update the system library file with the CDS parameter names. There should not be any errors.

It is a good idea on subsequent recompilations to compile without the -UL directive unless a new parameter name is purposely being added to the CDS. This guards against erroneous additions that might occur if a parameter name is accidentally mistyped while editing the file.

You can see all the parameters that have been defined by using File Manager Utilities to print the system library file &ASY>PARAMETR.SP. The file &ASY>SEGMENTS.SP can be printed to see all CDS file names that have been used. The -UL directive also controls whether CDS file names are entered into the library.

Compiling a CDS does not set aside storage for the parameter values; it simply defines the parameters to the system. The next required step is to build a point that uses the CDS parameters. Once a data point is built, the parameters of the CDS are part of the data point and are undifferentiated from other parameters of the data point.

At this point, you should back up the .SE and .SP files on &ASY and should also checkpoint the CG.

A.3 ACIDP/CRDP POINT BUILDING

CG point data can be recorded on Form *CG88-400* in preparation for the actual point-configuration process. Explanation of the entries is found in the *Computer Gateway Parameter Reference Dictionary* and the point-entry process is described in the *Data Entity Builder Manual.*

A brief outline of ACIDP/CRDP building follows:

1. From the Engineering Personality Main Menu, select the `COMPUTING MODULE` target in the Point Building column. This calls up the `CM BUILD AND CONFIGURATION MENU`.

2. Select the target appropriate to the point type to be built.

3. Enter the desired information into the `CM ACIDP/CRDP POINT ASSIGNMENT` display; save the point data in an IDF and then load the point.

If the point is a CRDP, it should have at least one associated "package," i.e., a CDS. For an ACIDP, any CDS is optional.

A.3.1 ACIDP Scheduling Recommendations

Set long RTPERIOD values for cyclic and periodic ACPs or set them to demand-only and determine how long they actually run before selecting the normal running period.

If the RTPERIOD in an ACIDP is short (close to the time required for the associated ACP to execute), it will be difficult or impossible to disconnect the ACIDP or to uninstall the ACP. If this happens, change the parameter INH_STAT to INHIBIT from the point's Detail Display at a Universal Station. Wait for the ACP to terminate as indicated by a change of Execution State to DELAY, then disconnect the ACP from the ACIDP.

HARDWARE INSTALLATION/CHECKOUT/SERVICE
Appendix B

This appendix contains hardware information useful in the installation, checkout, and service of the Host Processor-to-CG data link.

B.1 CG INSTALLATION, SERVICE AND TESTING

B.1.1 CG Installation

General information on site planning for and installation of TDC 3000^X LCN Nodes such as the CG is found in the *LCN Site Planning Manual*, the *LCN System Installation Manual*, and the *LCN System Checkout Manual*.

B.1.2 CG Service

General information on service for TDC 3000^X LCN Nodes such as the CG is found in the *Five/Ten-Slot Module Service* publication.

B.1.3 CG Testing

Like all LCN nodes, the CG contains built in firmware and software tests. Additional installation and off-line troubleshooting tests of the CG are provided by the Hardware Verification Test System (HVTS). Specific HVTS test programs that apply to the CG are

- Local Control Network Exerciser (LCNX)
- Memory Exerciser (MEMX)
- Communications Line Interface (CLIF)

See the *HVTS Reference Manual* for use information and restrictions.

Two of the Communications Line Interface board tests require special equipment. CLIF test 13 requires a loopback cable, and test 14 requires two loopback connectors.

For testing of data link transmissions in the normal (non-HVTS) operating environment, the use of a Data Analyzer is recommended. Typical of these devices are the AR 45000, the HP 4955A, and HP 4953A. See Figure 5-1 for a simulated data analyzer output of host/CG communication.

B.2 CABLES AND CONNECTORS

The following paragraphs summarize the pinning requirements for both RS-449 and RS-232C cables, including the test cables and connectors required by HVTS.

B.2.1 RS-449 CLI Adapter Board Connections

Figure B-1 illustrates the Line 1 and Line 2 cable connections at the CG's RS-449 Communications Line adapter board (part number 80360230-001).

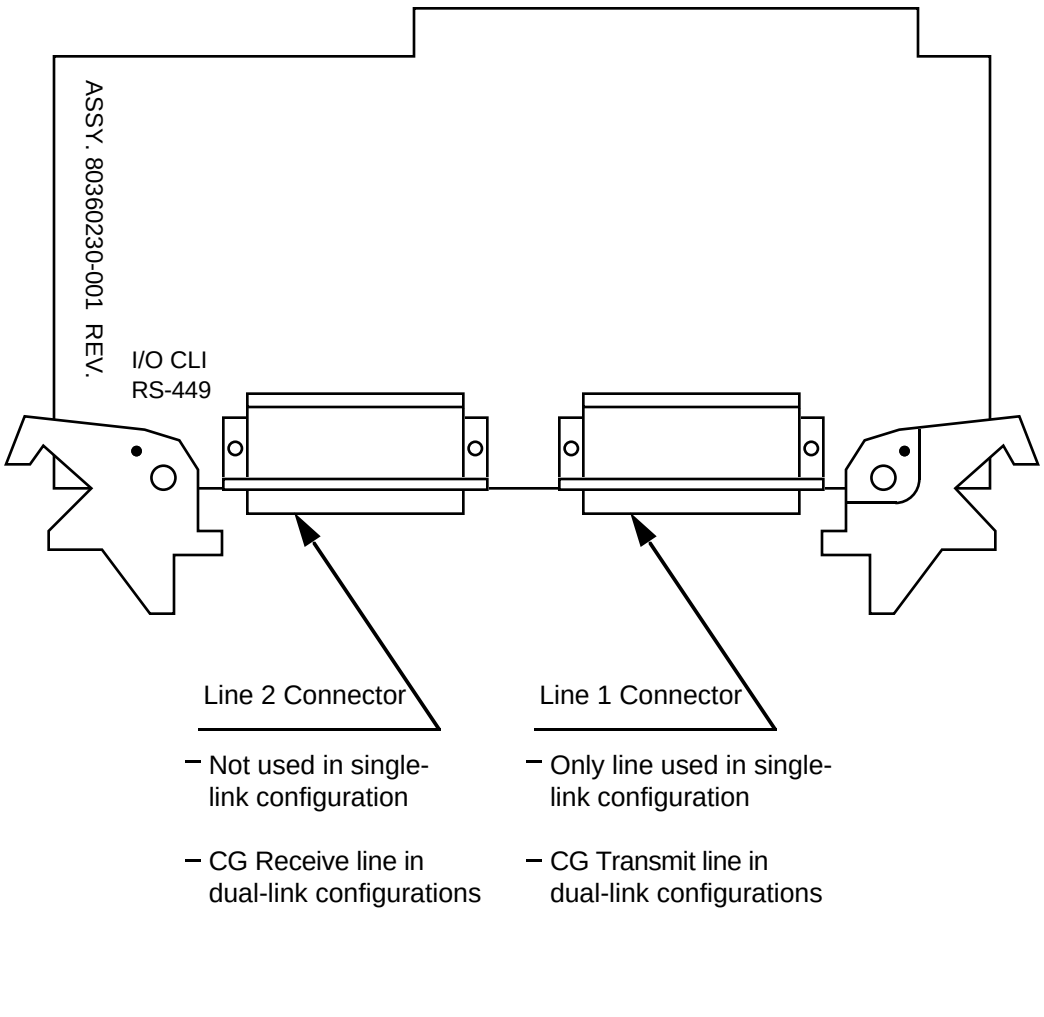
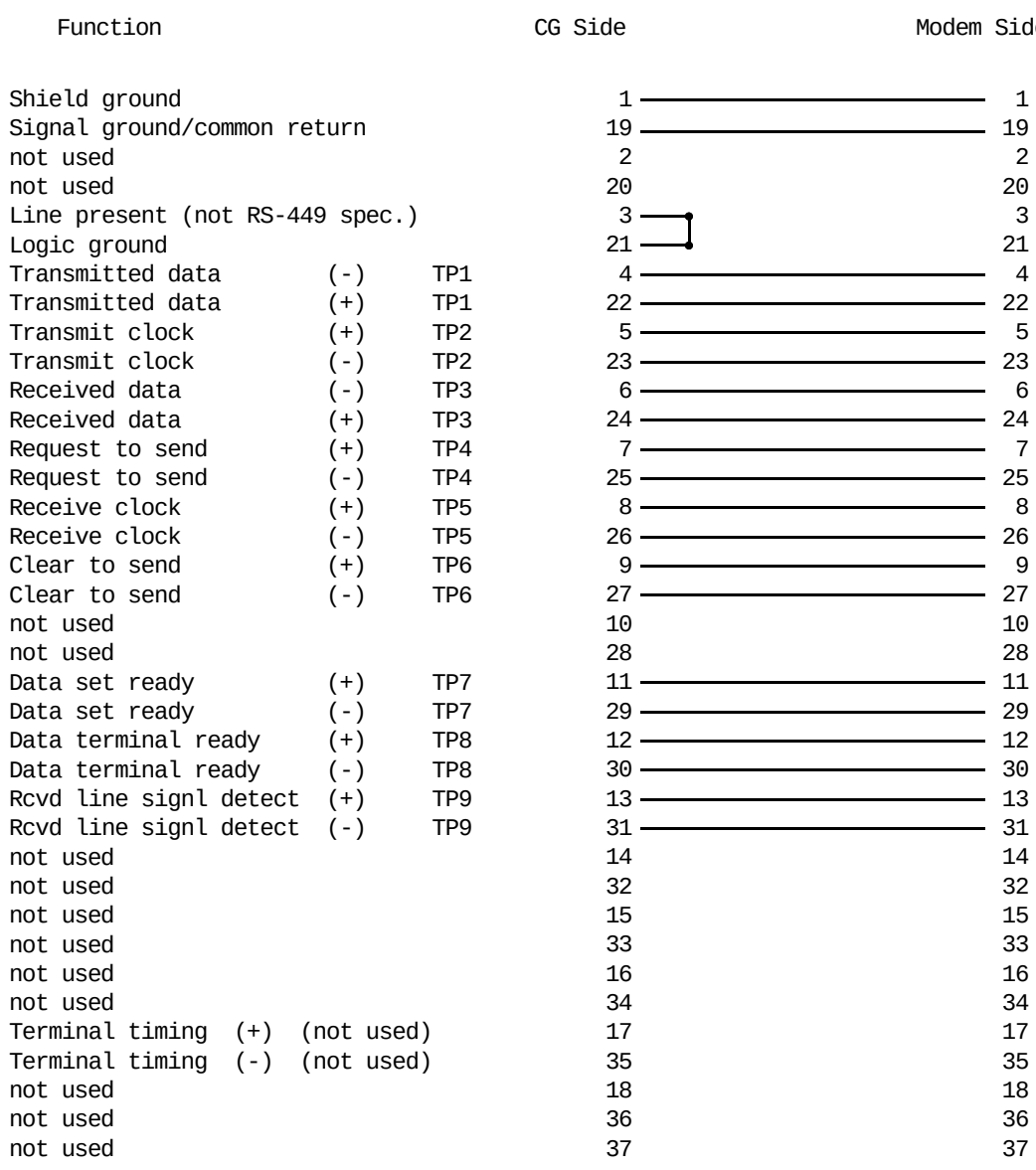


Figure B-1 — RS-449 CLI Adapter Board Connections 1563

B.2.2 RS-449 Standard (Modem Connection) Cable

- 15 meters, male-to-male connection
- Modem connected
- Both transmit and receive clocks are supplied by the modem
- RS-449 standard connector-pin assignment

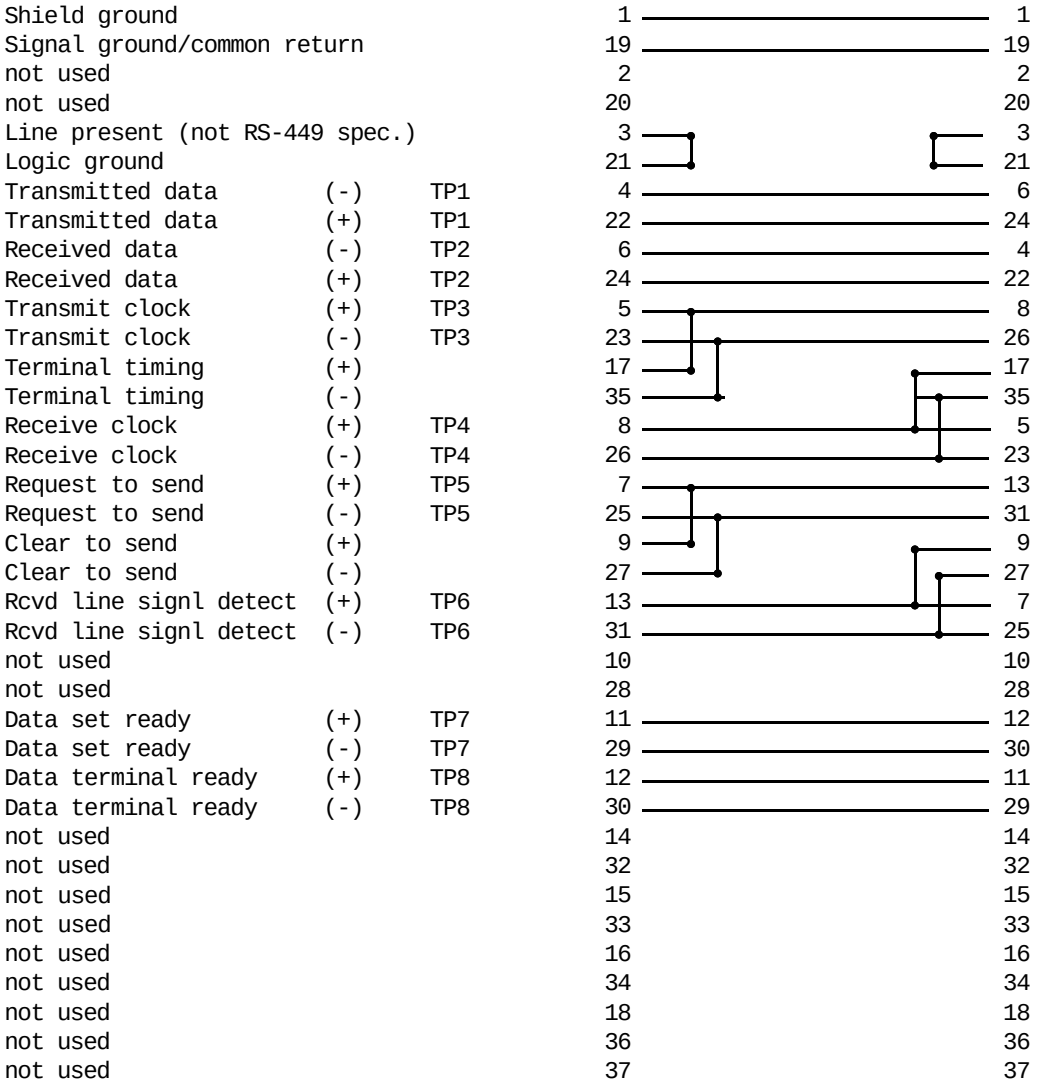


Note: TPn = Twisted pair line #n

Figure B-2 — RS-449 Standard (Modem) Cable 3221

B.2.3 RS-449 Direct Connect Cable

- 700 meters, male-to-male connection
- Direct connect
- Each data terminal (CG and host processor) supplies a transmit clock and a receive clock.

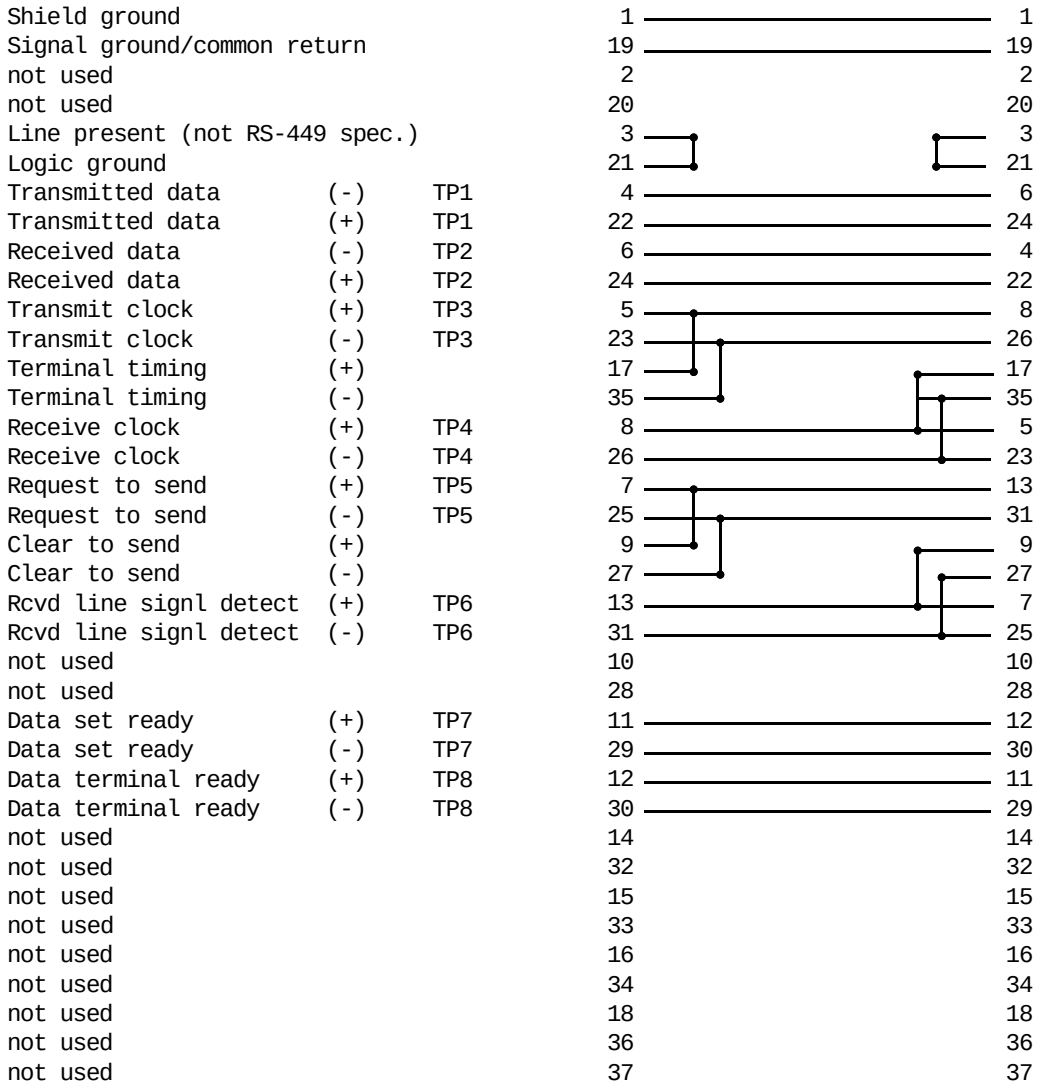


Note: TPn = Twisted pair line #n

Figure B-3 — RS-449 Direct Connect Cable 3222

B.2.4 RS-449 Loopback Cable

- Connects channels 0 to 3 and 2 to 1 for CLIF test 13
- Male-to-male connection



Function			CG Side	CG Side
Shield ground			1	1
Signal ground/common return			19	19
not used			2	2
not used			20	20
Line present (not RS-449 spec.)			3	3
Logic ground			21	21
Transmitted data	(-)	TP1	4	6
Transmitted data	(+)	TP1	22	24
Received data	(-)	TP2	6	4
Received data	(+)	TP2	24	22
Transmit clock	(+)	TP3	5	8
Transmit clock	(-)	TP3	23	26
Terminal timing	(+)		17	17
Terminal timing	(-)		35	35
Receive clock	(+)	TP4	8	5
Receive clock	(-)	TP4	26	23
Request to send	(+)	TP5	7	13
Request to send	(-)	TP5	25	31
Clear to send	(+)		9	9
Clear to send	(-)		27	27
Rcvd line signl detect	(+)	TP6	13	7
Rcvd line signl detect	(-)	TP6	31	25
not used			10	10
not used			28	28
Data set ready	(+)	TP7	11	12
Data set ready	(-)	TP7	29	30
Data terminal ready	(+)	TP8	12	11
Data terminal ready	(-)	TP8	30	29
not used			14	14
not used			32	32
not used			15	15
not used			33	33
not used			16	16
not used			34	34
not used			18	18
not used			36	36
not used			37	37

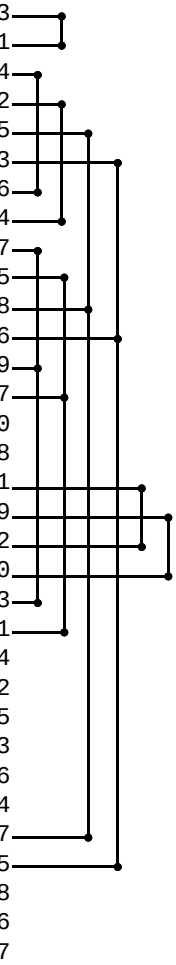
Note: TPn = Twisted pair line #n (TP8 not necessary to be twisted pair)

Figure B-4 — RS-449 Loopback Test Cable 3223

B.2.5 RS-449 Loopback Connector

- Male connector
- RS-449 standard connector pin assignments
- Connects channels 0 to 1 and 2 to 3 for CLIF test 14

Function			Connection in male connector
Shield ground			1
Signal ground/common return			19
not used			2
not used			20
Line present (not RS-449 spec.)			3
Logic ground			21
Transmitted data	(-)	TP1	4
Transmitted data	(+)	TP1	22
Transmit clock	(+)	TP2	5
Transmit clock	(-)	TP2	23
Received data	(-)	TP3	6
Received data	(+)	TP3	24
Request to send	(+)	TP4	7
Request to send	(-)	TP4	25
Receive clock	(+)	TP5	8
Receive clock	(-)	TP5	26
Clear to send	(+)	TP6	9
Clear to send	(-)	TP6	27
not used			10
not used			28
Data set ready	(+)	TP7	11
Data set ready	(-)	TP7	29
Data terminal ready	(+)		12
Data terminal ready	(-)		30
Rcvd line signl detect	(+)		13
Rcvd line signl detect	(-)		31
not used			14
not used			32
not used			15
not used			33
not used			16
not used			34
Terminal timing	(+)		17
Terminal timing	(-)		35
not used			18
not used			36
not used			37



Note: TPn = Twisted pair line #n

Figure B-5 — RS-449 Loopback Test Connector 3224

B.2.6 RS-232C CLI Adapter Board Connections

Figure B-6 illustrates the Line 1 and Line 2 cable connections at the CG's RS-232C Communications Line adapter board (part number 80360209-001).

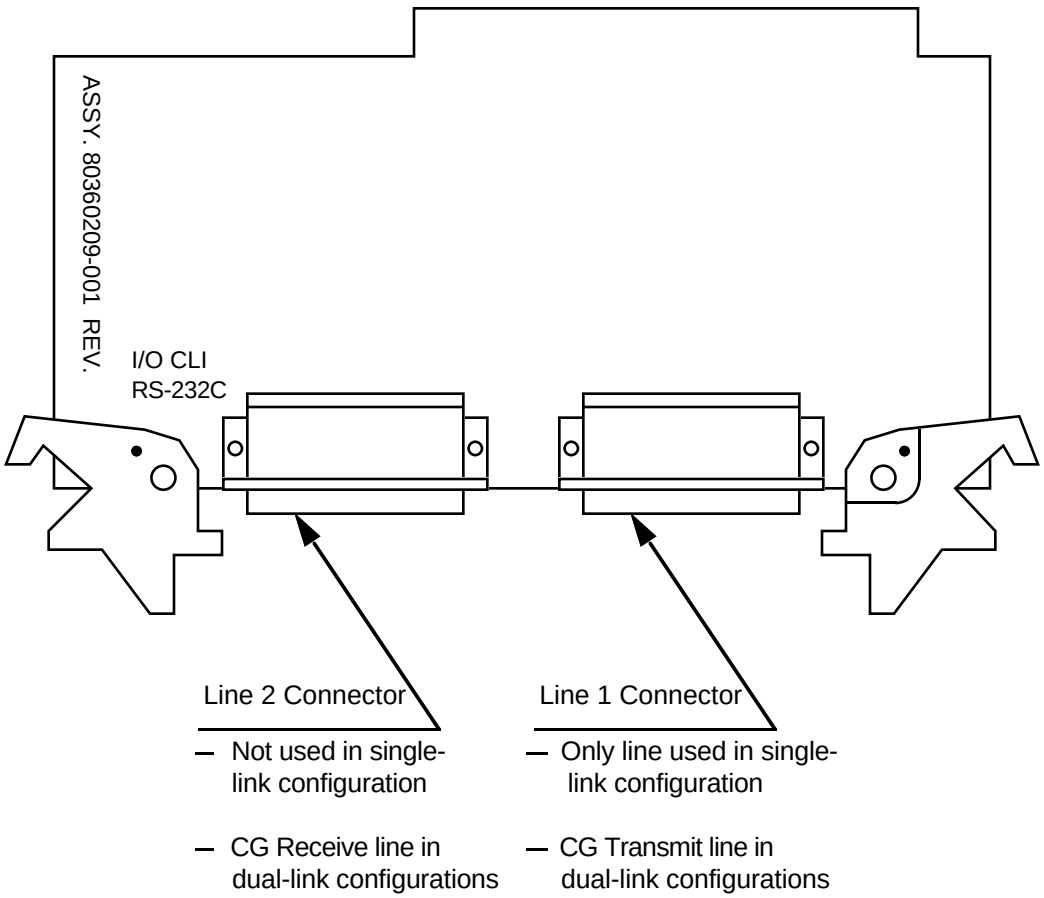


Figure B-6 — RS-232C CLI Adapter Board Connections 1564

B.2.7 RS-232C Standard (Modem Connection) Cable

- 15 meters, male-to-male connection
- Modem connected
- Both transmit and receive clocks are supplied by the modem
- RS-232C standard connector-pin assignment

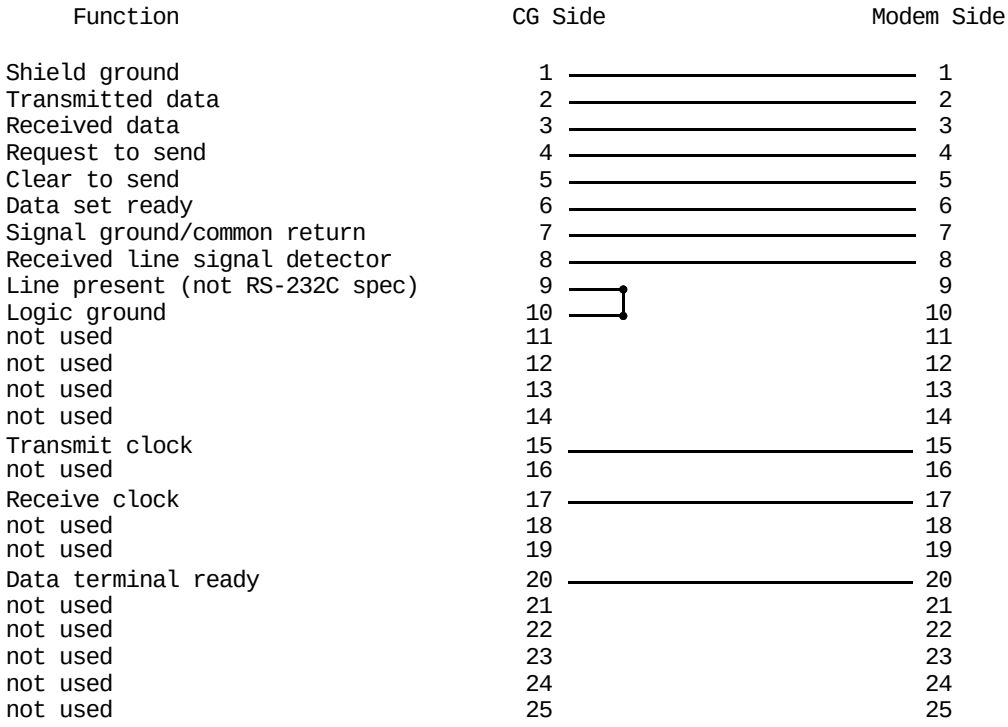


Figure B-7 — RS-232C Standard (Modem) Cable 3225

B.2.8 RS-232C Direct Connect Cable (Type I)

- 15 meters, male-to-male connection
- Direct connect
- Only one end (either CG or host processor) supplies the clock. If it is to supply the clock, the CG must be set in the direct connect mode.
- RS-232C standard connector-pin assignment

CAUTION

Because the two cable ends are pinned differently, care should be taken to position the cable correctly.

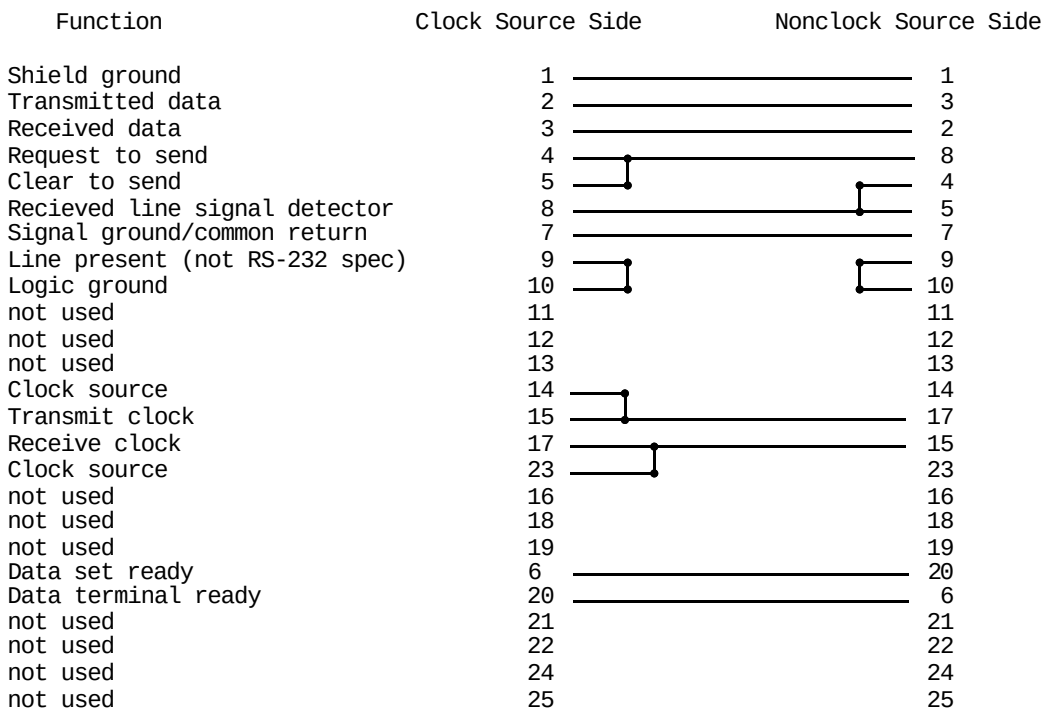


Figure B-8 — RS-232C Direct Connect Cable (Type I) 3226

B.2.9 RS-232C Direct Connect Cable (Type II)

- 15 meters, male-to-male connection
- Direct connect
- Each end supplies its own transmit clock and a receive clock.
- RS-232C standard connector-pin assignment

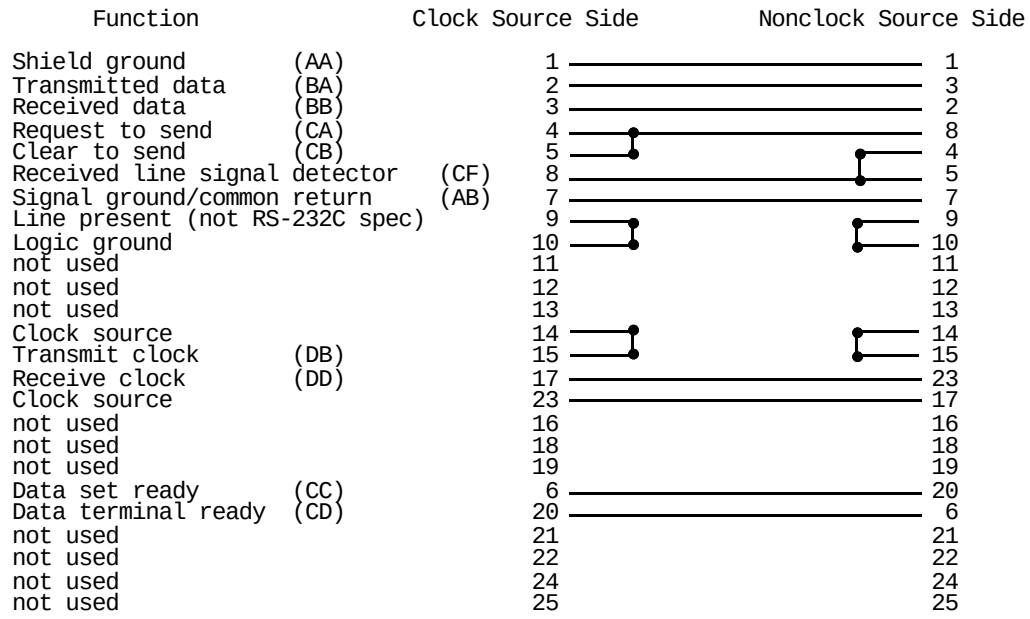


Figure B-9 — RS-232C Direct Connect Cable (Type II) 3227

B.2.10 RS-232C Loopback Cable

- Male-to-male connection
- Connects channels 0 to 3 and 2 to 1 for CLIF test 13

CAUTION

Because the two cable ends are pinned differently, care should be taken to position the cable correctly.

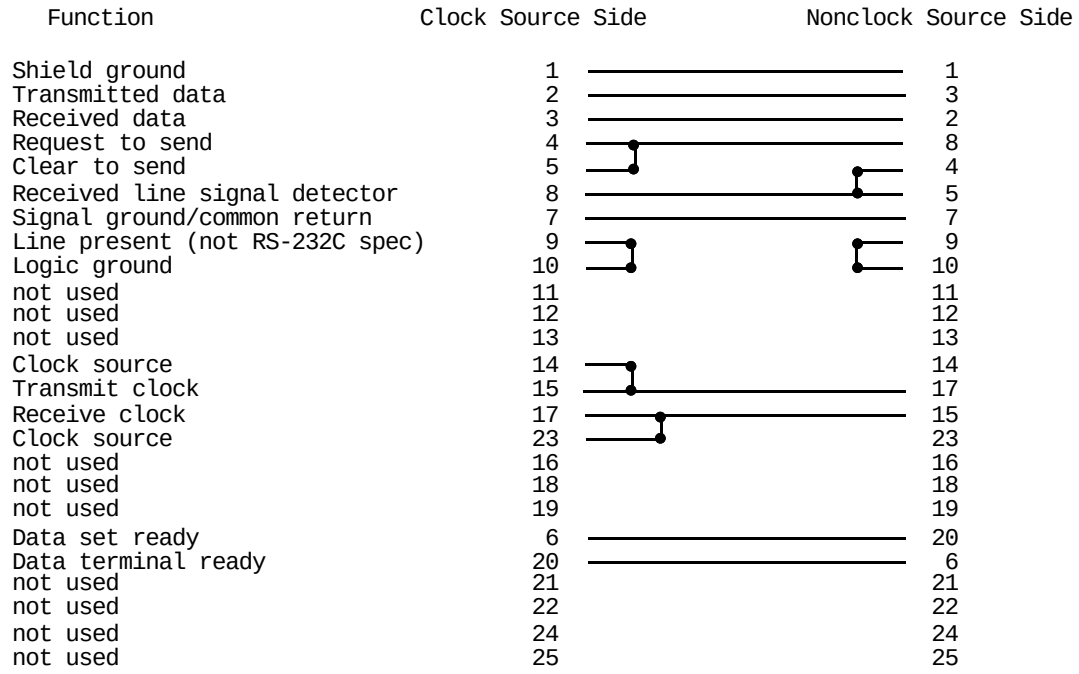


Figure B-10 — RS-232C Loopback Cable 3228

B.2.11 RS-232C Loopback Connector

- Male connector
- Connects channels 0 to 1 and 2 to 3 for CLIF test 14

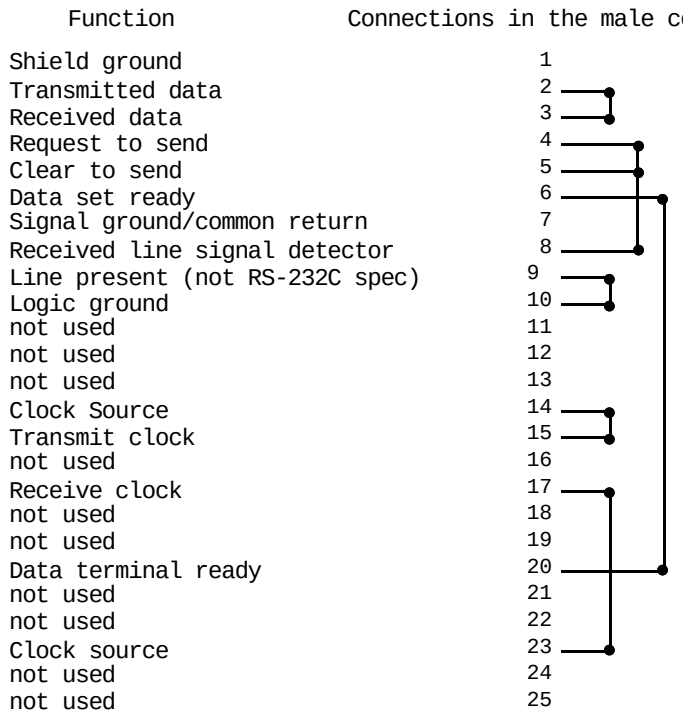
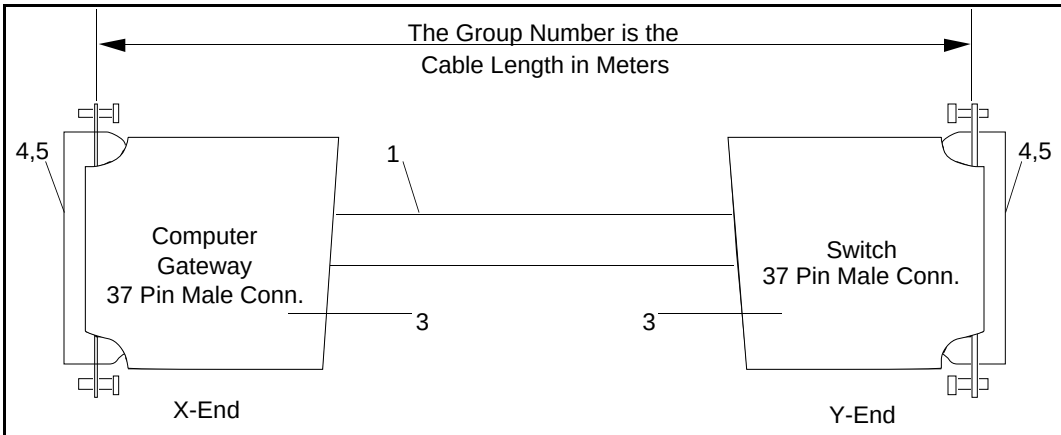


Figure B-11 — RS232C Loopback Connector 3229

B.2.12 CG to CG Switch Cable



Signal	Pin Number		Pin Number	Signal
Shield	1		1	Shield
Signal Ground	19 (Note 3)		19	Signal Ground
Cable Present	3 21		3 21	Cable Present
XMIT Data- XMIT Data+	4 22		4 22	XMIT Data- XMIT Data+
Recv. Data- Recv. Data+	6 24		6 24	Recv. Data- Recv. Data+
XMIT CLK- XMIT CLK+	5 23		5 23	XMIT CLK- XMIT CLK+
Recv. CLK - Recv. CLK +	8 26		8 26	Recv. CLK - Recv. CLK +
RTS + CTS +	7 9		7 9	RTS + CTS +
RTS - CTS -	25 27		25 27	RTS - CTS -
DCD + DCD -	13 31		13 31	DCD + DCD -
DTR + DTR -	12 30		12 30	DTR + DTR -
DSR + DSR -	11 29		11 29	DSR + DSR -

12770

Figure B-12 — CG to CG Switch Cable

B.2.12 CG to CG Switch Cable

Item No.	Part Number	Description	Group No. 001	Group No. 010	Group No. ###
1	51190868-110	Cable, 10 twist pair QTY=Length in meters	1	10	### Length in meters
2	51190831-030	Wire #24 AWG	AR	AR	AR
3	51190670-303	37 pin connector hood with screw retainers	2	2	2
4	51190670-204	37 pin connector plug	2	2	2
5	51190670-203	connector contact male pin	44	44	44

Note:
1. Maximum combined cable length of the two (2) cables in the system is 305 meters/1000 ft.
2. Connect both wires of a twisted pair to pin 19 at each end.
3. Jumpers use PL item 2 wire

B.2.13 VAX Cable to CG Switch

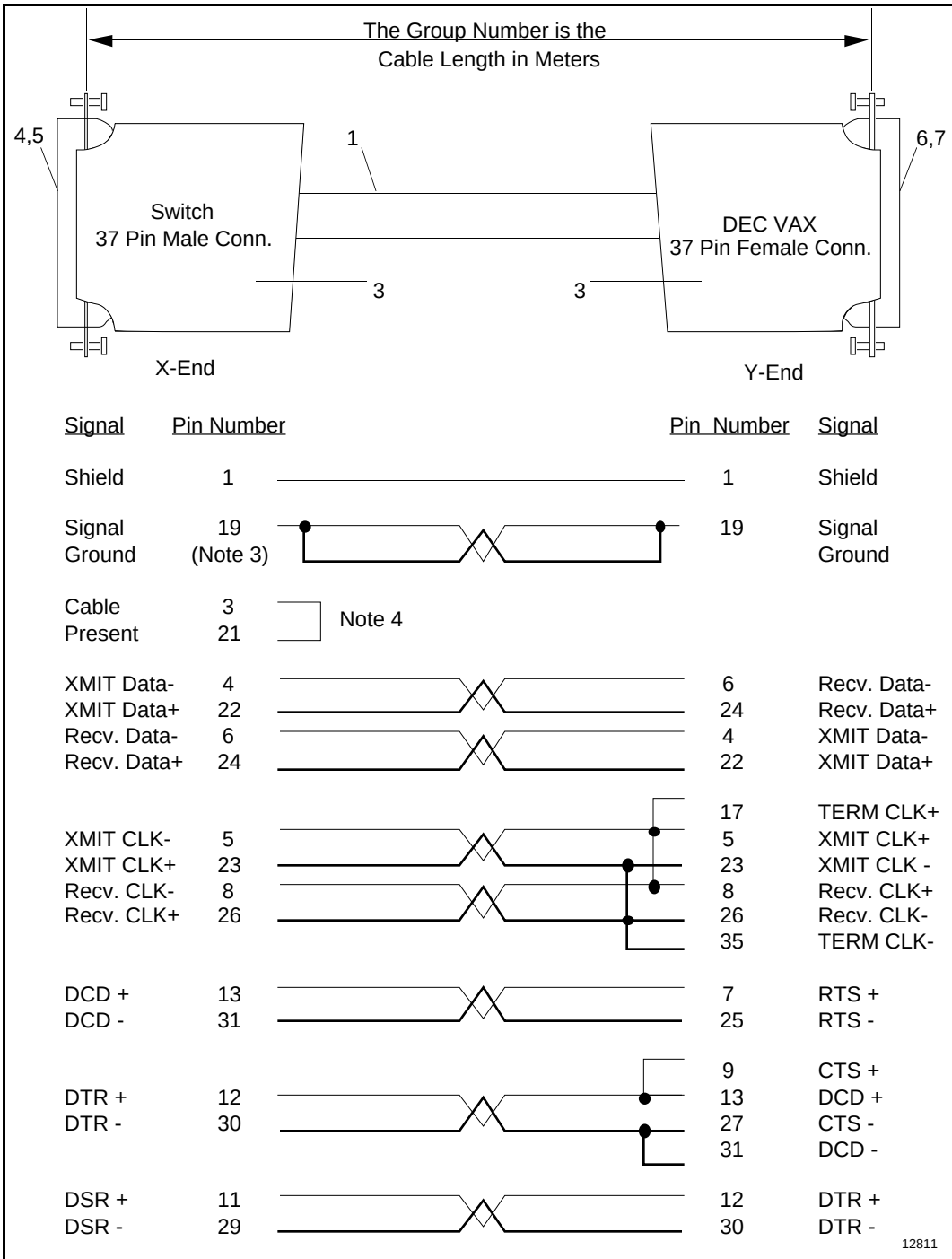


Figure B-13 — VAX Cable to CG Switch

B.2.13 VAX Cable to CG Switch

Item No.	Part Number	Description	Group No. 010	Group No. 100	Group No. ###
1	51190868-110	Cable, 10 twist pair QTY=Length in meters	10	100	### Length in meters
2	51190831-030	Wire #24 AWG	AR	AR	AR
3	51190670-303	37 pin connector hood with screw retainers	2	2	2
4	51190670-204	37 pin connector plug	1	1	1
5	51190670-203	Connector contact male pin	18	18	18
6	51190670-504	37 pin connector receptacle	1	1	1
7	51190670-403	Connector contact female socket	20	20	20

Note:
1. Maximum combined cable length of the two (2) cables in the system is 305 meters/1000 ft.
2. Connect both wires of a twisted pair to pin 19 at each end.
3. Jumpers use PL item 2 wire

TIMING DATA AND PERFORMANCE CONSIDERATIONS
Appendix C

This appendix contains information on performance and CG database size.

C.1 CG PROCESSING RESTRICTIONS

The host processor should limit each ACP to one request message at a time.

The CG allows the host processor to send multiple messages and can stack a maximum of 20 messages for processing. Actual concurrent processing, however, is limited by space limitations and by limitations on specific message types. Examples of these limitations are Continuous History 4, Single Point Get/Store 1, Multi-point Get/Store 4, Miscellaneous Functions 1, and Build Internal DDT 1.

If the CG message-receiving capacity is exceeded, it sends the expected response message with a return status indicating that its queues are full.

C.2 CG PERFORMANCE LIMITS

When transferring data by Internal DDT, it is important to realize that there is an overhead associated with processing each request. It is, therefore, inefficient to use Internal DDTs with small numbers of points. Where throughput is important, Honeywell recommends that nonhistory Internal DDTs should contain at least 200 point-parameter definitions. If the points come from multiple LCN nodes, the best use is made of TDC 3000^{X}'s parallel processing capabilities and best performance is realized.

Following are some of the limits on the CG's ability to share information with other nodes on the LCN:

Response to other nodes for ACIDP or CRDP values—45 parameters per second.

Get or Store of values to other nodes (table driven at 300 parameters per Internal DDT)—1.5 DDTs per second (of type Real).

Get or Store of Whole Array—1000 parameters from another node in 2 seconds.

Get History—80 'points' per minute (assuming 60 values per point for one minute snapshots).

CG performance may be affected by other loads placed on its information sources and destinations (HG, AM, HM), as well as loads placed on itself for display of ACIDP/CRDP information, activation of ACPs, etc; therefore, these figures must be considered approximate. It is recommended that users update a cluster-loading analysis when a CG is added to their system.

Get History performance has more variables; however, the essential thing to remember is that the difference in time required for a larger set of points (within the limit of 5986 data words) is relatively small.

Throughput may be enhanced by using multiple ACPs to request the acquisition of history data because the CG can support up to four concurrent history requests.

C.3 INPUT DATA PRECOLLECTION

The input data precollection feature enables collection of multipoint data values very close in time to a "trigger" event, and reduces the elapsed time required to complete a get data request (measured from the time an ACP is triggered at the CG until the data has been moved to the host processor).

Precollection requires that a Internal DDT (resident in the CG) is connected to an ACIDP-ACP pair, and that one or more of its triggers (schedule, operator demand, and point process special) are set ON. When a trigger occurs, data collection for those data points specified by the DDT is started in parallel with the sending of an ACP turn on message to the host processor, rather than waiting on a get data request to be initiated by the ACP.

Precollected data waits in CG memory for the first get data request from an attached ACP. There is no indication in the get data response of whether the data being returned was precollected or collected following the get data request.

For additional information see the following message types:

- Internal DDT Connect or Disconnect Request (paragraph 4.4.7)
- Store Internal DDT in CG Request (paragraph 4.5.3)
- ACIDP Execution State Change Request (paragraph 4.4.6)

The CG allows the same Internal DDT to be connected to more than one ACIDP-ACP pair; however, this practice is not recommended because there is the possibility that one ACP could get time dependent data intended for another.

ASSIGNMENT OF PROCESS UNITS TO CG
Appendix D

This appendix contains information relating to the assignment of process units to the CG.

1. The number of process units that can be assigned to a CG is 63 minus the number of CGs assigned to the same checkpoint volume. For example, if you have two CGs assigned to one HM for checkpointing, there can be no more than 61 units assigned.

2. Because the CG node status display shows only points that are assigned to its area and to the CG node, you can never see more than 36 (area limit).

3. ACP access to parameters in other LCN nodes, both read and write, is independent of which units are assigned to the CG. That is, it does not matter to which unit the ACP's ACIDP is assigned.

4. Event Initiated Processing from the HG and CL Messages from the AM and MC are independent of process unit assignment in the CG.

5. The operator demand of an ACP from its ACIDP detail display requires that the ACIDP is in a unit assigned to that Universal Station's area, or that the US is in Engineer keylock level. The same limitation applies to any CG point-parameter that you may wish to store to, from a custom display.

6. An ACP can send an operator message to only the Unit its ACIDP is assigned to. Thus, only areas with that unit assigned will receive the message.

7. A process unit can be assigned to only one CG. If there are multiple CGs on the LCN, each must have a unique assignment of process units.

To summarize, for most flexibility, all units assigned to CGs should be assigned to all areas that need to communicate with that CG. The units that contain the points that are accessed by the CG do not have to be assigned to the CG.

In most cases, there is no real need to assign more than one unit to a CG.

Index

Index

Index

Index

Index

Index

Index

READER COMMENTS

Honeywell IAC Automation College welcomes your comments and suggestions to improve future editions of this and other publications.

You can communicate your thoughts to us by fax, mail, or toll-free telephone call. We would like to acknowledge your comments; please include your complete name and address.

BY FAX: Use this form and fax to us at (602) 313-4108.

BY TELEPHONE: In the U.S.A., use our toll-free number 1*800-822-7673 (available in the 48 contiguous states except Arizona; in Arizona dial 1-602-313-5558).

BY MAIL: Use this form; detach, fold, tape closed, and mail to us.

Title of Publication: **Computer Gateway User Manual** Issue Date: **4/96**

Publication Number: **CG11-410**

Writer: **J. Danner**

COMMENTS: _____

_____

_____

_____

_____

_____

_____

_____

RECOMMENDATIONS:_____

_____

_____

_____

_____

_____

_____

_____

NAME _____ DATE _____

TITLE _____

COMPANY _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

TELEPHONE _____ FAX _____

(If returning by mail, please tape closed; Postal regulations prohibit use of staples.)

Communications concerning technical publications should be directed to:

Automation College
Industrial Automation and Control
Honeywell Inc.
2820 West Kelton Lane
Phoenix, Arizona 85023-3028

FOLD FOLD

From: _____

NO POSTAGE
NECESSARY
IF MAILED
IN THE USA

BUSINESS REPLY MAIL

FIRST CLASS **PERMIT NO. 4332** PHOENIX, ARIZONA

POSTAGE WILL BE PAID BY ....

Honeywell

Industrial Automation and Control
2820 West Kelton Lane
Phoenix, Arizona 85023-3028

Attention: Manager, Quality

FOLD FOLD

Cut Along Line

Additional Comments:

Honeywell

Industrial Automation and Control
Honeywell Inc.
16404 North Black Canyon Highway
Phoenix, Arizona 85023-3033

Helping You Control Your World