StorageTek®

PN 313471401

OS 2200 Client System Component (CSC)

4R1 Installation Guide

July 2001

THIS DOCUMENT DOES NOT EXTEND OR CREATE WARRANTIES OF ANY NATURE, EITHER EXPRESSED OR IMPLIED. Storage Technology Corporation cannot accept any responsibility for use of the information in this document or of any associated software program. Storage Technology Corporation assumes no responsibility for any data corruption or erasure as a result of the information in this document, including any associated software program. You are responsible for backing up your data. You should be careful to ensure that use of the information complies with all applicable laws, rules, and regulations of the jurisdictions with respect to which it is used.

Information in this document is subject to change, and Storage Technology Corporation reserves the right to revise or modify the information in this document without prior notification. In the event of changes, this document will be revised. Contact Storage Technology Corporation to verify that you have the most current version of this document.

Comments concerning the information in this document should be directed to:

Storage Technology Corporation
2270 South 88th Street
Louisville, CO 80028-4232
(800) 678-4430

CONTENTS

PREFACE

1. QUICK START

2. CONFIGURING THE COMMUNICATION METHOD

3. USING COMUS TO BUILD AND INSTALL CSC

4. USING SOLAR TO INSTALL CSC

5. COMPLETING THE INSTALLATION PROCESS

6. STARTING CSC & OPTIONAL VERIFICATIONS

INDEX

EFFECTIVE PAGES

TABLES

FIGURES

PREFACE

PURPOSE

This is the *OS 2200 Client System Component (CSC) Installation Guide.* This guide describes how to install the CSC 4R1 product group on the Unisys 1100/2200 Client System (the “client”). CSC is the software used by the client to communicate with the Solaris®-based Library Control System or the Nearline Control Solution (the “server”) and the Automated Cartridge System (ACS).

AUDIENCE

This guide is written for *experienced* systems support staff and operators. It assumes that you are familiar with the following hardware and software:

- ACS
- Unisys Series 1100 and 2200 computers
- Solaris-based Library Control System
- Nearline Control Solution
- Executive Control Language (ECL)
- SOLAR and/or COMUS

HOW TO USE THIS DOCUMENT

Chapter 1. Quick Start

This chapter summarizes the basic information you need to know to install and configure CSC. It does not provide detailed installation information, and is intended *only* for experienced users.

Chapter 2. Configuring the Communication Method

This chapter describes how to configure the communication method.

Chapter 3. Using COMUS to Build and Install CSC

This chapter describes how to register, build, and install the CSC product group. The generation uses standard Unisys COMUS REGISTER, BUILD, and INSTALL procedures. Examples of the COMUS dialogue provide details for each step of the process.

Chapter 4. Using SOLAR to Install CSC

This chapter describes how to register and install CSC using SOLAR.

Chapter 5. Completing the Installation Process

This chapter describes the few remaining steps that must be considered before the CSC installation is complete.

Chapter 6. Starting CSC & Optional Verifications

This chapter describes how to start CSC. In addition, it describes optional procedures you can use to verify that CDI is correctly configured and operational.

Back Matter

The manual includes an index.

COMMAND SYNTAX NOTATION

This manual uses the following conventions for representing command syntax notation and message displays:

UPPERCASE	indicates a command or keyword
lowercase italic	indicates a user- or system-supplied variable value. For example, in XX=*userid*, you enter the actual userid for *userid*.
abbreviation	indicates a command that can be abbreviated to its minimum acceptable form. For example, ENAble can be abbreviated to ENA.
vertical bar I	separates operand alternatives. For example, A I B indicates that you must select either A or B.
brackets []	indicate an option that can be omitted. For example, [A I B I C] indicates that you can select A, B, C, or nothing.
braces {}	indicate an option that you *must* choose. For example, {A I B} indicates that you must choose either A or B.
underlining	indicates the system default. If you do not enter a parameter or value, the system will supply the underscored value. For example, A I B I C indicates that if you do not choose an option, the system will default to C.
ellipses ...	indicate that entries can be repeated as often as necessary.
SMALLCAPS	indicate a key, such as XMIT or F1.

RELATED DOCUMENTATION

OS 2200 Client System Component (CSC) Technical Bulletin, Unisys Corporation (7844 8693)

OS 2200 Client System Component (CSC) System Administrator's Guide, Unisys Corporation (7844 8685)

OS 2200 Client System Component (CSC) Operations Guide, Unisys Corporation (7844 8644)

OS 2200 Client System Component User Interface (CSCUI) Programmer's Reference Manual, Unisys Corporation (7844 8677)

OS 2200 Client System Component (CSC) Client Direct Interconnect (CDI) Troubleshooting Guide, Unisys Corporation (7844 8651)

Automated Cartridge System Library Software Product Document Set for Solaris, Storage Technology Corporation.

Nearline Control Solution 4.0 Publication Kit, Storage Technology Corporation (313456301)

Exec System Software Operations Reference Manual, Unisys Corporation (7831 0281)

Executive Control Language (ECL) and FURPUR Reference Manual, Unisys Corporation (7830 7949)

COMUS End User Reference Manual, Unisys Corporation (7830 7758)

Software Library Administrator (SOLAR) End User Reference Manual, Unisys Corporation (7831 06*05*)

Exec System Software Executive Requests Programming Reference Manual, Unisys Corporation (7830 7899)

Communications Management System (CMS1100) Configuration Reference Manual, Unisys Corporation (7830 9846)

ClearPath HMP IX Series Cooperative Processing Communications Platform (CPCOMM) Configuration and Operations Guide, Unisys Corporation (7844 8438)

1. QUICK START

This chapter summarizes, in quick, readable terms, the basic information you need to know to configure your communication method and install the CSC product group. This information is intended *only* for experienced users, and to provide a high-level overview of the process.

1.1 QUICK START PROCEDURES

If you are an experienced user, you may want to use the following general procedures.

NOTE

CSC 4R1 is compatible with either the COMUS or SOLAR method of installation. This chapter uses the COMUS method to illustrate all register, build, and installation processes.

1.1.1 Before You Install the CSC Product Group

1. Review the contents of your release package.

   The CSC 4R1 release package contains the CSC product tape and documentation. You use the CSC product tape to build and install the CSC product group.

2. Review your site's bank descriptor index (BDI) requirements.

   Your site will require one BDI for the Client Direct Interconnect (CDI), if selected, and two BDIs for the CSC User Interface (CSCUI), if selected

3. Obtain the latest Interim Correction.

   There may be an update to this level of the CSC product. You can obtain this from the Unisys Web based support page or from your Unisys Support Representative.

1.1.2 Configure CDI, if Necessary

To configure CDI, you must perform the following steps:

1. Configure the CPAs using OS 2200 configuration statements.

2. Set the CPA channel addresses.

3. Connect the CPA(s) to the client and LAN.

4. Set the appropriate channel type in the CPA.

5. Configure CDI network SGSs in the CDI$PARAM element in the SI file of the CSC product tape.

1.1.3 Build and Install CSC

Once you've established and configured the communication environment, you're ready to build and install the CSC product group:

1. Perform a COMUS REGISTER of the CSC product tape.

2. Perform a COMUS BUILD of the CSC product group, applying the latest Interim Correction as an update.

3. Perform a COMUS INSTALL of the CSC product group.

4. Complete the installation process:
   - Verify the CSC parameter elements CSC$PARAM and CSC$DRIVE, and update if necessary. Note that the Solaris-based Library Control System is the default. If you are converting to the Nearline Control Solution then you must update CSC$PARAM.
   - If you opt to use the CSC User Interface (CSCUI), you must configure the CSCUI exits.
   - If you opt to use CDI, you must verify the CDI$PARAM element and update if necessary.
   - Configure NODE statements for input to the EXEC generation, if CDI is configured.

1.1.4 Start CSC

Once you've completed the previous steps, you can start CSC. As a reminder, you must reload common banks if you have upgraded, reconfigured, or want to initialize your environment. **Note that data loss may result if you reload the CSCUI common banks between CSC executions.**

1. If you're using CSCUI, reload the CSCUICBA and CSCUICBB common banks.

2. If CDI is configured, START the CDI run.

3. START the CSC run.

1.1.5 Perform Optional Verifications

If you are using CDI and want to perform optional verifications to determine that it is functioning properly, follow these steps:

1. Run the CPATST utility.

2. Execute the *CDI PING command.

2. CONFIGURING THE COMMUNICATION METHOD

This chapter describes how to configure the communication method for your system.

2.1 OVERVIEW

CSC supports the following interfaces for communication with the server:

CDI
CMS
CPCOMM

CSC can be configured to use any combination of these interfaces. The CSC configuration also defines paths through the various LAN attachment hardware (CPA, HLC, DCP, ICA) used by each interface. CSC uses only one path at a time for sending requests to the server.

This chapter presents the following information:

- Configuring CDI
- Configuring CMS
- Configuring CPCOMM
- Overview of configuring CSC communications
- Sample configurations

2.2 CONFIGURING CDI

To use CDI on your system, you should follow these steps:

Configure and install CPAs.

Create or update CDI network configuration element, CDI$PARAM.

Create or update the INTERFACE and PATH statements in the CSC$PARAM configuration element. See the “Overview” section in this chapter and the “Statement Details” in Chapter 5.

2.2.1 Configure and Install CPAs

2.2.1.1 Configure CPAs in OS 2200

First, you must configure all CPAs in OS 2200. Configure each CPA as an arbitrary control unit, with three arbitrary devices, using the syntax:

```
NODE cpa-name IS ARBCU AND CONNECTS TO ;
   channel-name VIA SUBCHANNEL n
NODE arbdev-name1,arbdev-name2,arbdev-name3 ARE ;
   ARBDEV and CONNECT TO cpa-name VIA  ;
   DEVICE-ADDRESS 0,1,2
```

Table 2-1. OS 2200 CPA Configuration Variables

Variable	Description
*arbdev-name*n	Names for the three arbitrary devices internal to the CPA, where *n* represents device 1, 2, or 3.
channel-name	Channel to which the CPA is attached.
cpa-name	Name of the CPA control unit.
n	Subchannel number.

Following is an example of the OS 2200 configuration statements for *CPA0* in a configuration with one CPA:

```
NODE CPACU0 IS ARBCU AND CONNECTS TO BMC0  ;
   VIA SUB-CHANNEL 0
NODE CPAD00,CPAD01,CPAD02 ARE ARBDEV AND  ;
   CONNECT TO CPACU0 VIA DEVICE-ADDRESS 0,1,2
```

2.2.1.2 SET CPA CHANNEL ADDRESSES

See the *4400 ACS CPA Technical Guide* for details.

2.2.1.3 Set Appropriate Channel Type

See the *4400 ACS CPA Technical Guide* for details.

If you install CPAs on a 2200/400 system or on certain 2200/600 systems, then the streaming mode setting of the CPA may need updating. This is described in the *CDI Troubleshooting Guide*.

2.2.1.4 Connect CPA(s) to UNISYS Client and LAN

This should be performed by a Unisys Customer Service Engineer (CSE).

2.2.2 CDI Network Configuration Element, CDI$PARAM

The CDI network configuration resides in an element named CDI$PARAM. If CSC is not already installed, then this element is taken from the SI file of the CSC product tape. This CDI$PARAM element contains default IP addresses for a single client, single LAN, single server configuration. You must configure a CDI$PARAM element for each client in the network. This element contains the following types of statements:

```
CLIENT client-name ''UNISYS-2200''
CPA cpa-name ADDR ip-addr DEVICE dev-0, dev-1, dev-2
```

```
KEYIN keyin-name
SEC_LEVEL_x keyin-name
SERVER IP ADDR ip-addr
ROUTE IP ADDR ip-addr VIA cpa-name
FORWARD ROUTE FOR ip-addr0 IS ip-addr1 VIA cpa-name
DEFAULT ROUTE IS ip-addr VIA cpa-name
```

Each of the six lines represents a different Stream Generation Statement (SGS). The following sections describe each of these statements. For examples using these statements in combination with CSC configuration statements, refer to the “Sample Configurations” section of this chapter.

2.2.2.1 KEYIN SGS

The KEYIN SGS is optional. It allows a site to change the console keyin used by CDI. This SGS has the following syntax:

```
KEYIN keyin-name
```

where *keyin-name* is the console keyin to which CDI will respond.

2.2.2.2 SEC_LEVEL_x SGS

The SEC_LEVEL_x SGS is optional. It allows a site to designate the demand terminal security level required for each of the CDI console commands. A full description of this SGS parameter is contained in the “Changing Security Levels” section of Chapter 5.

2.2.2.3 CLIENT SGS

The CLIENT SGS is required. It defines the name of the client system used within CSC. This SGS has the following syntax:

```
CLIENT client-name ''UNISYS-2200''
```

where *client-name* represents the name of the local host client, and can contain up to six characters.

2.2.2.4 CPA SGS

The CPA SGS is required. It defines a CPA that CDI can use. You should have one CPA SGS for each configured CPA. This SGS has the following syntax:

```
CPA cpa-name ADDR ip-addr DEVICE dev-0, dev-1, dev-2
```

Table 2-2. CPA SGS Parameters

Parameter	Description
cpa-name	The six-character CPA name used within CSC. This name will be used in *CDI console keyins.
Ip-addr	Internet Protocol address of this CPA. This address uses commas (,) instead of dots (.). For example, the definition of address 192.168.0.3 is 192,168,0,3.
dev-0, dev-1, dev-2	Names of the OS 2200 devices for one CPA. D*ev-0*, *dev-1* and *dev-2 correspond to arbdev-name1, arbdev-name2, and arbdev-name3 of the NODE statement in the OS 2200 configuration.*

2.2.2.5 SERVER IP ADDR SGS

Each SERVER IP ADDR SGS defines one IP address on the server. You should use one SERVER IP ADDR SGS for each server LAN adapter through which CSC can communicate. This SGS has the following syntax:

```
SERVER IP ADDR ip-addr
```

where *ip-addr* is the Internet address of the specified server connection.

NOTE

Each SERVER IP ADDR SGS must have a corresponding IPADDR statement in the CSC$PARAM element.

2.2.2.6 ROUTE IP ADDR SGS

The ROUTE IP ADDR SGS defines a path from a CPA to a specific server LAN adapter. You should use one ROUTE IP ADDR SGS for each directly connected combination of CPA and server LAN adapter. This statement can be used only when the CPA and the server LAN adapter are on the same subnet. Routing code within CDI allows ROUTE IP ADDR statements from multiple CPAs to a single server LAN adapter. Routing on the server does not support paths from multiple server LAN adapters to a single CPA. This SGS has the following syntax:

```
ROUTE IP ADDR ip-addr VIA cpa-name
```

Table 2-3. ROUTE IP ADDR SGS Parameters

Parameter	Description
Ip-addr	The Internet address of the server LAN adapter. There must be a SERVER IP ADDR statement for this address.
cpa-name	The CPA through which packets will be sent to this server address. There must be a CPA statement with a matching *cpa-name.*

NOTE

Use the FORWARD ROUTE SGS instead of the ROUTE IP ADDR SGS if the CPA and server LAN adapter are on different subnets.

Example

The following is a typical CDI$PARAM element for a configuration of one CPA communicating with one server LAN adapter. The OS 2200 DEVICE configuration is CPAD00, CPAD01, CPAD02 for the CPA. The server IP address is 192.168.0.1, and the client CPA address is 192.168.0.3

```
CLIENT U22001 ''Unisys-2200/412''
CPA ALPHA ADDR 192,168,0,3 DEVICE CPAD00,CPAD01,CPAD02
SERVER IP ADDR 192,168,0,1
ROUTE IP ADDR 192,168,0,1 VIA ALPHA
```

2.2.2.7 FORWARD ROUTE SGS

The FORWARD ROUTE SGS defines a path from a CPA through a router to a specific server LAN adapter. This statement should be used when the CPA and the server LAN adapter are not on the same subnet. This FORWARD ROUTE statement also tells CDI the IP address of the router. Routing code within CDI allows FORWARD ROUTE statements from multiple CPAs to a single server LAN adapter. Routing on the server does not support paths from multiple server LAN adapters to a single CPA. This SGS has the following syntax:

```
FORWARD ROUTE FOR ip-addr0 IS ip-addr1 VIA cpa-name
```

Table 2-4. FORWARD ROUTE SGS Parameters

Parameter	Description
Ip-addr0	The Internet address of the server LAN adapter. There must be a SERVER IP ADDR statement for this address.
Ip-addr1	The Internet address of the router to the destination *ip-addr0.*
cpa-name	The CPA through which packets will be sent to server *ip-addr0.* There must be a CPA statement with a matching *cpa-name.*

NOTE

Use the ROUTE IP ADDR SGS instead of the FORWARD ROUTE SGS if the CPA and server LAN adapter are on the same subnet.

Example

The following is a typical CDI$PARAM element for a configuration with a CPA on one subnet connected by a router to a server LAN adapter on another subnet. The OS 2200 DEVICE configuration is CPAD10, CPAD11, CPAD12 for the CPA. The server IP address is 192.168.0.1, the router IP address is 192.168.1.254, and the client CPA address is 192.168.1.3.

```
CLIENT U22001 ''Unisys-2200/412''
CPA BRAVO ADDR 192,168,1,3 DEVICE CPAD10,CPAD11,CPAD12
SERVER IP ADDR 192,168,0,1
FORWARD ROUTE FOR 192,168,0,1 IS 192,168,1,254 VIA BRAVO
```

2.2.2.8 DEFAULT ROUTE SGS

The DEFAULT ROUTE SGS defines the network route to use if the destination IP address is not in the same subnet as the source IP address. This SGS has the following syntax:

```
DEFAULT ROUTE IS ip-addr VIA cpa-name
```

In most configurations, the ROUTE IP ADDR or FORWARD ROUTE statement should be used instead of the DEFAULT ROUTE statement. Those statements define host routes to specific server addresses rather than the network route that is internally calculated for this statement. Check with your Network Administrator before using the DEFAULT ROUTE statement.

Table 2-5. DEFAULT ROUTE SGS Parameters

Parameter	Description
Ip-addr	The Internet address of the server LAN adapter. This will be ANDed with the subnet mask to form a network route.
cpa-name	To send a packet to *ip-addr0*, route the packet to *ip-addr1* VIA this cpa. This *cpa-name* should match one of the CPA SGS *cpa-names.*

2.2.3 Update CSC$PARAM

The CSC configuration element, CSC$PARAM, must be updated to tell that the CDI interface is being used and to define the paths through CDI to the server. There is an overview of the required statements later in this chapter. The full description of these statements is in Chapter 5.

2.3 CONFIGURING CMS

Before CSC can use CMS as an interface to the server, a process must be configured in CMS for use by CSC. Information on configuring processes within CMS can be found in the *CMS 1100 Installation and Configuration Guide in the section titled* "Configuring Processes That Use the TCP/IP TSAM Interface."

Following are sample statements from a CMS configuration element to define a TSAM process that can be used by CSC:

```
PROCESS,CSC     TYPE,TSAM  ;
                PASSWORD,CSCPW  ;
                INTERNET-ADR,IAETH1
```

The CSC configuration element, CSC$PARAM, must be updated to tell that the CMS interface is being used and to define the paths through CMS to the server. There is an overview of the required statements later in this chapter. The full description of these statements is in Chapter 5.

Following are sample PATH and INTERFACE statements from a CSC$PARAM configuration element, to use the process defined above:

```
Interface CMS1 uses CMSA,CSC,CSCPW
Path ALPHA uses CMS1 connects 192,168,0,7 to 192,168,0,1
```

NOTE

Your site ip-addresses must be used in the CSC$PARAM configuration. See your site administrator for the correct ip-addresses to use in the configuration.

2.4 CONFIGURING CPCOMM

Before CSC can use CPCOMM as an interface to the server, a process must be configured in CPCOMM for use by CSC. Information on configuring processes within CPCOMM can be found in the *Cooperative Processing Communications Platform (CPCOMM) Configuration and Operations Guide*

Following are sample statements from a CPCOMM configuration element to define a process that can be used by CSC and its linkage to an adapter IP address:

```
PROCESS,CSC     PASSWORD,CSCPW
CA,CA001
LINK,L001       CA,CA001  ETHERNET-LAN,LAN001
IP,IP001        LINK,L001 IP-ADDRESS,192.168.0.8
```

The CSC configuration element, CSC$PARAM, must be updated to tell that the CPCOMM interface is being used and to define the paths through CPCOMM to the server. There is an overview of the required statements later in this chapter. The full description of these statements is in chapter 5.

Following are sample PATH and INTERFACE statements from a CSC$PARAM configuration element, to use the process defined above:

```
Interface CPCOMMA uses CPCOMMA,CSC,CSCPW
Path DELTA uses CPCOMMA connects 192.168,0,8 to 192,168,0,1
```

2.5 OVERVIEW OF CSC COMMUNICATION CONFIGURATION

The CSC configuration must be updated to use the paths defined by the communication configurations earlier in this chapter. This section provides an overview of the required CSC configuration statements and how they interact with the communication interface configuration. Refer to Chapter 5 for the syntax of these CSC configuration statements.

2.5.1 Define Interface Programs

Each INTERFACE statement defines a program that CSC can use for communicating with the server. One or more INTERFACE statements are required for CSC operation. The general format of an INTERFACE statement is

```
INTERFACE intf-name USES intf-program[,options]
```

where *intf-name* is the name used in CSC commands for this interface. *Intf-program* is either CDI or the installation mode for CSC or CPCOMM. *Options* are specific to each interface program.

Following are typical INTERFACE statements.

```
INTERFACE INTFA USES CDI,0400410,0141000
INTERFACE INTFB USES CMSA,CSC,CSC
```

2.5.2 Define Paths to the Server

Each PATH statement defines and names a communication path between CSC and the server. It specifies only the end points of the connection. One end point is a server LAN address. The other is LAN interface hardware controlled by the INTERFACE program (CPA, HLC, DCP, etc). Any routing for the path is specified in the interface configuration statements.

The order of path statements is significant. When CSC selects a path, it searches paths in the order of the PATH statements in CSC$PARAM. Your path statements should occur in the order of preference for use. The general format of a PATH statement is:

```
PATH path-name USES intf-name CONNECTS ca-addr TO
server-addr
```

where *path-name* is the name of the path used for CSC commands; *intf-name* is the name of the interface; *ca-addr* is the address of the CPA, HLC, DCP, or ICA; and *server-addr* is the address of the server LAN adapter.

2.6 SAMPLE CONFIGURATION SCHEME AND STATEMENTS

2.6.1 CDI with One CPA / One LAN / One Server Address

Figure 2-1 shows the connection of one client to one server through one LAN. One CDI connection is established through a single CPA.

Client System	ALPHA Internet Address	Server (LAN Adapter) Address
1	*192.168.0.3*	*192.168.0.1*

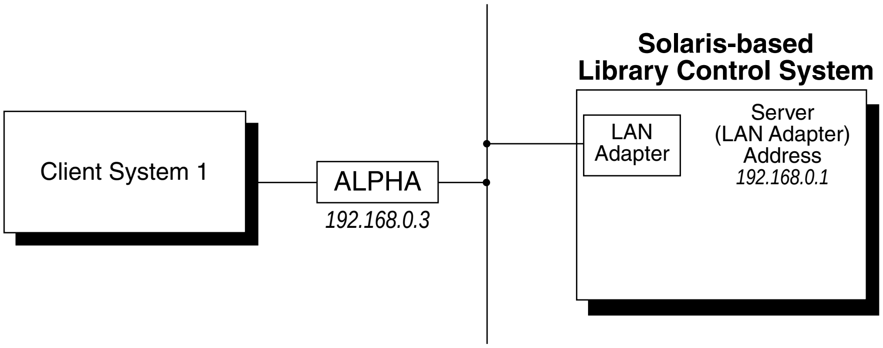


Figure 2-1. CDI / One CPA / One LAN / One Server Address

2.6.1.1 Configuration Statements

OS 2200 Configuration for ALPHA

```
NODE CAPCU0 IS ARBCU AND CONNECTS TO BMC0 VIA SUB-CHANNEL 0
NOTE CPAD00,CAPD01,CPAD02 ARE ARBDEV AND ;
   CONNECT TO CPACU0 VIA DEVICE-ADDRESS 0,1,2
```

*CDI Network Configuration (SYS$LIB$*CSC-PARM.CDI$PARAM)*

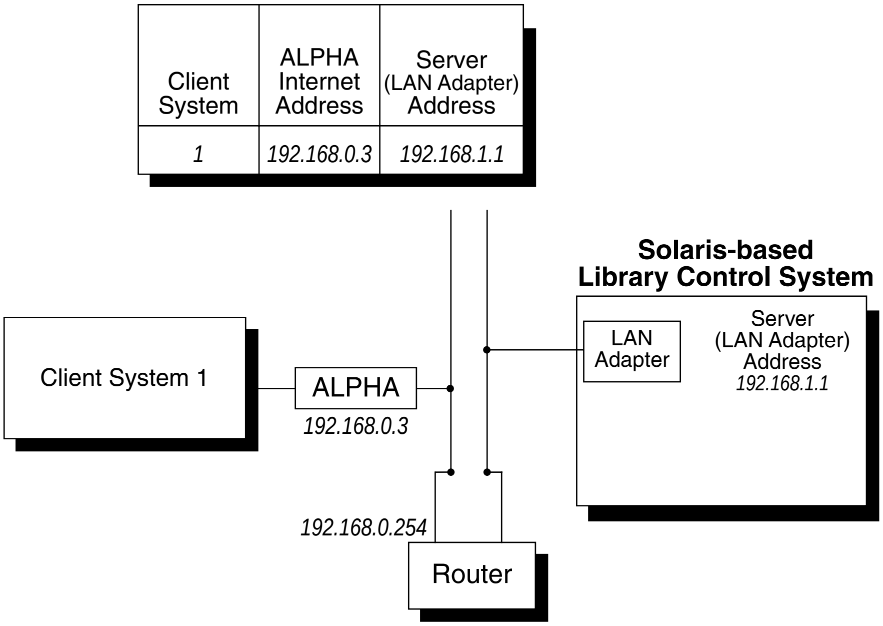
```
CLIENT U22001 ' 'Unisys-2200/412''
CPA ALPHA ADDR 192,168,0,3 DEVICE CPAD00,CPAD01,CPAD02
SERVER IP ADDR 192,168,0,1
ROUTE IP ADDR 192,168,0,1 VIA ALPHA
```

*Relevant CSC configuration (SYS$LIB$*CSC-PARM.CSC$PARAM)*

```
INTERFACE CDIINT USES CDI
PATH ALPHA USES CDIINT CONNECTS 192,168,0,3 TO 192,168,0,1
```

2.6.2 CDI with One CPA / Routed LAN / One Server Address

Figure 2-2 shows the connection of one client to one server through multiple LANs. One CDI connection is established through a single CPA.



Client System	ALPHA Internet Address	Server (LAN Adapter) Address
1	*192.168.0.3*	*192.168.1.1*

Figure 2-2. CDI / One CPA / Routed LAN / One Server Address

2.6.2.1 Configuration Statements

OS 2200 Configuration for ALPHA

```
NODE CAPCU0 IS ARBCU AND CONNECTS TO BMC0 VIA SUB-CHANNEL 0
NOTE CPAD00,CAPD01,CPAD02 ARE ARBDEV AND ;
   CONNECT TO CPACU0 VIA DEVICE-ADDRESS 0,1,2
```

*CDI Network Configuration (SYS$LIB$*CSC-PARM.CDI$PARAM)*

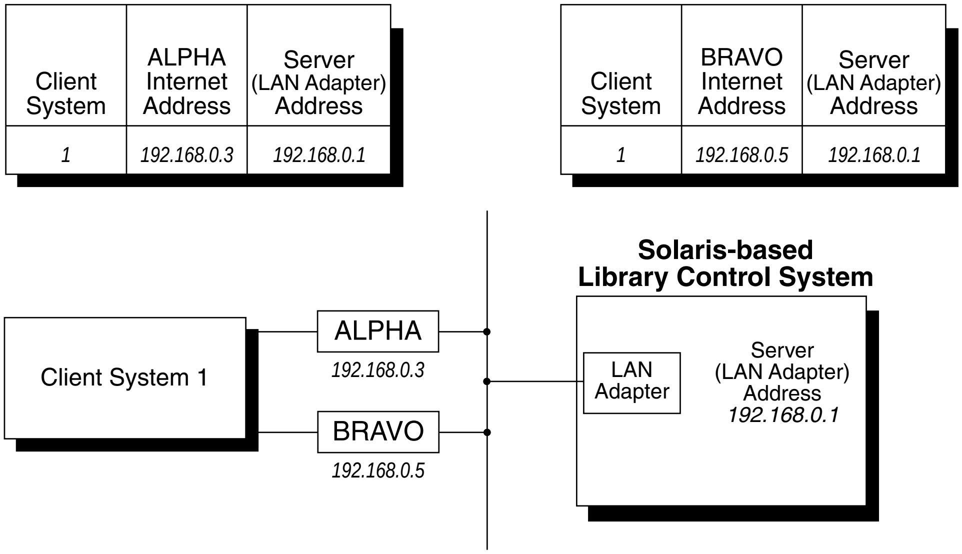
```
CLIENT U22001 ' 'Unisys-2200/412''
CPA ALPHA ADDR 192,168,0,3 DEVICE CPAD00,CPAD01,CPAD02
SERVER IP ADDR 192,168,1,1
FORWARD ROUTE FOR 192,168,1,1 IS 192,168,0,254 VIA ALPHA
```

*Relevant CSC configuration (SYS$LIB$*CSC-PARM.CSC$PARAM)*

```
INTERFACE CDIINT USES CDI
PATH ALPHA USES CDIINT CONNECTS 192,168,0,3 TO 192,168,1,1
```

2.6.3 CDI with Two CPAs / One LAN / One Server Address

Figure 2-3 shows the connection of one client to one server through one LAN. One CDI connection is established through a single CPA.



Client System	ALPHA Internet Address	Server (LAN Adapter) Address
1	*192.168.0.3*	*192.168.0.1*

Client System	BRAVO Internet Address	Server (LAN Adapter) Address
1	*192.168.0.5*	*192.168.0.1*

Figure 2-3. CDI / Two CPAs / One LAN / One Server Address

2.6.3.1 Configuration Statements

OS 2200 Configuration for ALPHA

```
NODE CPACU0 IS ARBCU AND CONNECTS TO BMC0 VIA SUB-CHANNEL 0
NODE CPAD00,CPAD01,CPAD02 ARE ARBDEV AND  ;
   CONNECT TO CPACU0 VIA DEVICE-ADDRESS 0,1,2
```

OS 2200 Configuration for BRAVO

```
NODE CPACU1 IS ARBCU AND CONNECTS TO BMC1 VIA SUB-CHANNEL 0
NODE CPAD10,CPAD11,CPAD12 ARE ARBDEV AND  ;
   CONNECT TO CPACU1 VIA DEVICE-ADDRESS 0,1,2
```

*CDI Network Configuration (SYS$LIB$*CSC-PARM.CDI$PARAM)*

```
CLIENT U22001 ''Unisys-2200/500''
CPA ALPHA ADDR 192,168,0,3 DEVICE CPAD00,CPAD01,CPAD02
CPA BRAVO ADDR 192,168,0,5 DEVICE CPAD10,CPAD11,CPAD12
SERVER IP ADDR 192,168,0,1
ROUTE IP ADDR 192,168,0,1 VIA ALPHA
ROUTE IP ADDR 192,168,0,1 VIA BRAVO
```

*Relevant CSC configuration (SYS$LIB$*CSC-PARM.CSC$PARAM)*

```
INTERFACE CDIINT USES CDI
PATH ALPHA USES CDIINT CONNECTS 192,168,0,3 TO 192,168,0,1
PATH BRAVO USES CDIINT CONNECTS 192,168,0,5 TO 192,168,0,1
```

2.6.4 CDI with Two CPAs / Two LANs / Two Server Addresses

Figure 2-4 has one client, with two CPAs, interacting with one server through dual LANs. One CDI connection is established through each CPA.

Client System	ALPHA Internet Address	Server (LAN Adapter) Address
1	*192.168.0.3*	*192.168.0.1*

Client System	BRAVO Internet Address	Server (LAN Adapter) Address
1	*192.168.1.3*	*192.168.1.1*

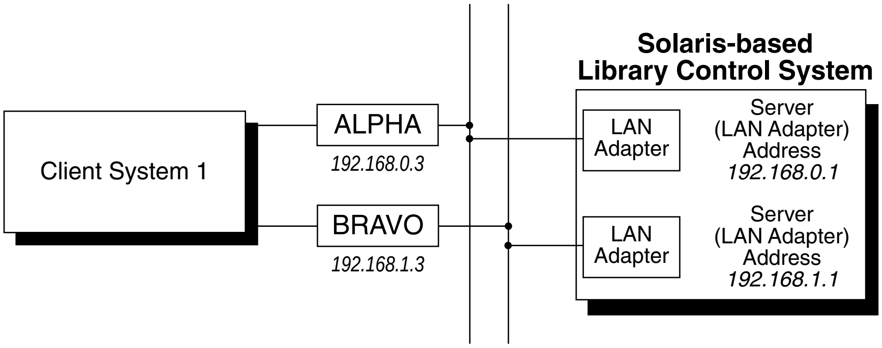


Figure 2-4. CDI / Two CPAs / Dual LANs / Two Server Addresses

2.6.4.1 Configuration Statements

OS 1100 Configuration for ALPHA

```
NODE CPACU0 IS ARBCU AND CONNECTS TO BMC0 VIA SUB-CHANNEL 0
NODE CPAD00,CPAD01,CPAD02 ARE ARBDEV AND  ;
   CONNECT TO CPACU0 VIA DEVICE-ADDRESS 0,1,2
```

OS 1100 Configuration for BRAVO

```
NODE CPACU1 IS ARBCU AND CONNECTS TO BMC1 VIA SUB-CHANNEL 0
NODE CPAD10,CPAD11,CPAD12 ARE ARBDEV AND  ;
   CONNECT TO CPACU1 VIA DEVICE-ADDRESS 0,1,2
```

*CDI Network Configuration (SYS$LIB$*CSC-PARM.CDI$PARAM)*

```
CLIENT U22001 ''Unisys-2200/500''
CPA ALPHA ADDR 192,168,0,3 DEVICE CPAD00,CPAD01,CPAD02
CPA BRAVO ADDR 192,168,1,3 DEVICE CPAD10,CPAD11,CPAD12
SERVER IP ADDR 192,168,0,1
SERVER IP ADDR 192,168,1,1
ROUTE IP ADDR 192,168,0,1 VIA ALPHA
ROUTE IP ADDR 192,168,1,1 VIA BRAVO
```

*Relevant CSC configuration (SYS$LIB$*CSC-PARM.CSC$PARAM)*

```
INTERFACE CDIINT USES CDI
PATH ALPHA USES CDIINT CONNECTS 192,168,0,3 TO 192,168,0,1
PATH BRAVO USES CDIINT CONNECTS 192,168,1,3 TO 192,168,1,1
```

2.6.5 CDI with Two CPAs / Multiple LANs / Two Server Addresses

Figure 2-5 has one client, with two CPAs, interacting with one server through multiple LANs. One CDI connection is established through each CPA.

Client System	ALPHA Internet Address	Server (LAN Adapter) Address
1	192.168.0.3	192.168.2.1

Client System	BRAVO Internet Address	Server (LAN Adapter) Address
1	192.168.1.3	192.168.1.1

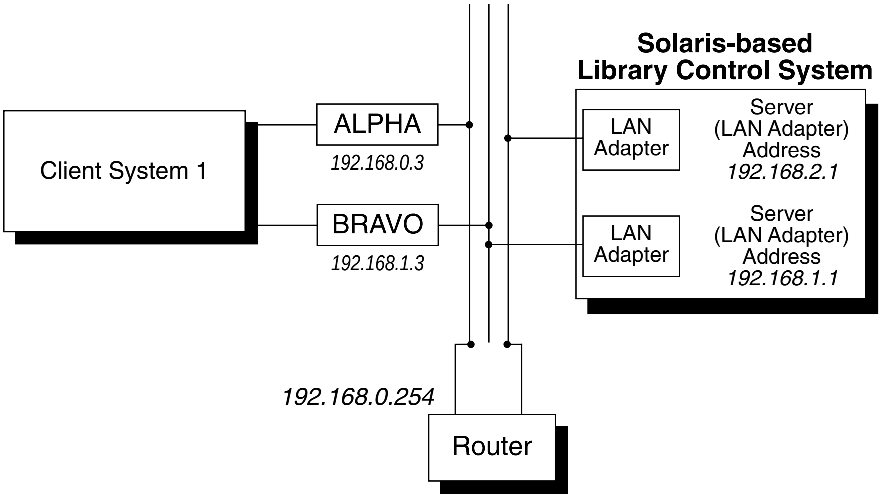


Figure 2-5. CDI / Two CPAs / Multiple LANs / Two Server Addresses

2.6.5.1 Configuration Statements

OS 1100 Configuration for ALPHA

```
NODE CPACU0 IS ARBCU AND CONNECTS TO BMC0 VIA SUB-CHANNEL 0
NODE CPAD00,CPAD01,CPAD02 ARE ARBDEV AND  ;
   CONNECT TO CPACU0 VIA DEVICE-ADDRESS 0,1,2
```

OS 1100 Configuration for BRAVO

```
NODE CPACU1 IS ARBCU AND CONNECTS TO BMC1 VIA SUB-CHANNEL 0
NODE CPAD10,CPAD11,CPAD12 ARE ARBDEV AND  ;
   CONNECT TO CPACU1 VIA DEVICE-ADDRESS 0,1,2
```

*CDI Network Configuration (SYS$LIB$*CSC-PARM.CDI$PARAM)*

```
CLIENT U22001 ''Unisys-2200/500''
CPA ALPHA ADDR 192,168,0,3 DEVICE CPAD00,CPAD01,CPAD02
CPA BRAVO ADDR 192,168,1,3 DEVICE CPAD10,CPAD11,CPAD12
SERVER IP ADDR 192,168,2,1
SERVER IP ADDR 192,168,1,1
ROUTE IP ADDR 192,168,1,1 VIA BRAVO
FORWARD ROUTE FOR 192,168,2,1 IS 192,168,0,254 VIA ALPHA
```

*Relevant CSC configuration (SYS$LIB$*CSC-PARM.CSC$PARAM)*

```
INTERFACE CDIINT USES CDI
PATH ALPHA USES CDIINT CONNECTS 192,168,0,3 TO 192,168,2,1
PATH BRAVO USES CDIINT CONNECTS 192,168,1,3 TO 192,168,1,1
```

2.6.6 CMS with One Adapter / One LAN / One Server Address

Figure **2**-6 shows the connection of one client to one server through CMS.

Client System	GAMMA	Server (LAN Adapter) Address
1	*192.168.0.7*	*192.168.0.1*

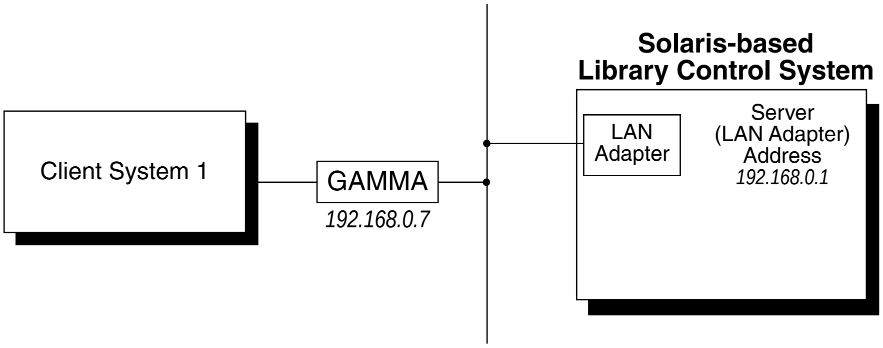


Figure 2-6. One Client/TSAM/One LAN

2.6.6.1 Configuration Statements

CMS configuration for a TSAM process for CSC

```
PROCESS,CSC    TYPE,TSAM  ;
PASSWORD,CSCPW  ;
INTERNET-ADR,IAETH1
```

*Relevant CSC configuration (SYS$LIB$*CSC-PARM.CSC$PARAM)*

```
INTERFACE CMSINT USES CMSA,CSC,CSCPW
PATH GAMMA USES CMSINT CONNECTS 192,168,0,7 TO 192,168,0,1
```

2.6.7 CDI and CMS with One Adapter Each / One LAN / One Server Address

Figure 2-7 shows one client using either CDI or CMS to connect to one server through one LAN. CDI uses one CPA. CMS uses one LAN adapter (HLC, DCP, etc). Due to the order of the PATH statements the preferred path is via CMS.

Client System	ALPHA Internet Address	Server (LAN Adapter) Address
1	*192.168.0.3*	*192.168.0.1*

Client System	GAMMA Internet Address	Server (LAN Adapter) Address
1	*192.168.0.7*	*192.168.0.1*

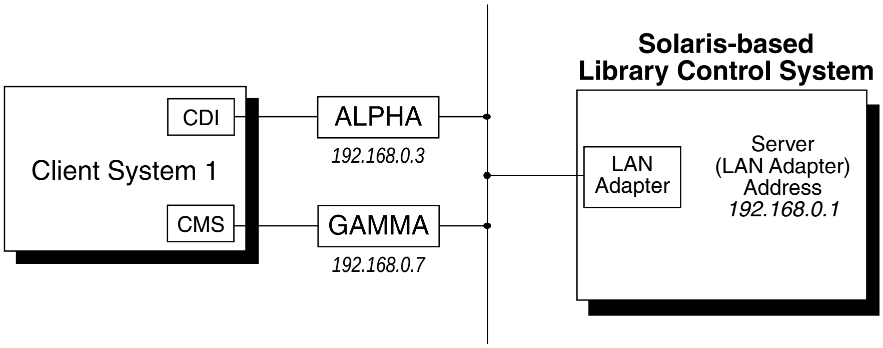


Figure 2-7. CDI / Two CPAs / One LAN / One Server Address

2.6.7.1 Configuration Statements

OS 2200 Configuration for ALPHA

```
NODE CPACU0 IS ARBCU AND CONNECTS TO BMC0 VIA SUB-CHANNEL 0
NODE CPAD00,CPAD01,CPAD02 ARE ARBDEV AND  ;
   CONNECT TO CPACU0 VIA DEVICE-ADDRESS 0,1,2
```

*CDI Network Configuration (SYS$LIB$*CSC-PARM.CDI$PARAM)*

```
CLIENT U22001 ''Unisys-2200/500''
CPA ALPHA ADDR 192,168,0,3 DEVICE CPAD00,CPAD01,CPAD02
SERVER IP ADDR 192,168,0,1
ROUTE IP ADDR 192,168,0,1 VIA ALPHA
```

CMS configuration for a TSAM process for CSC

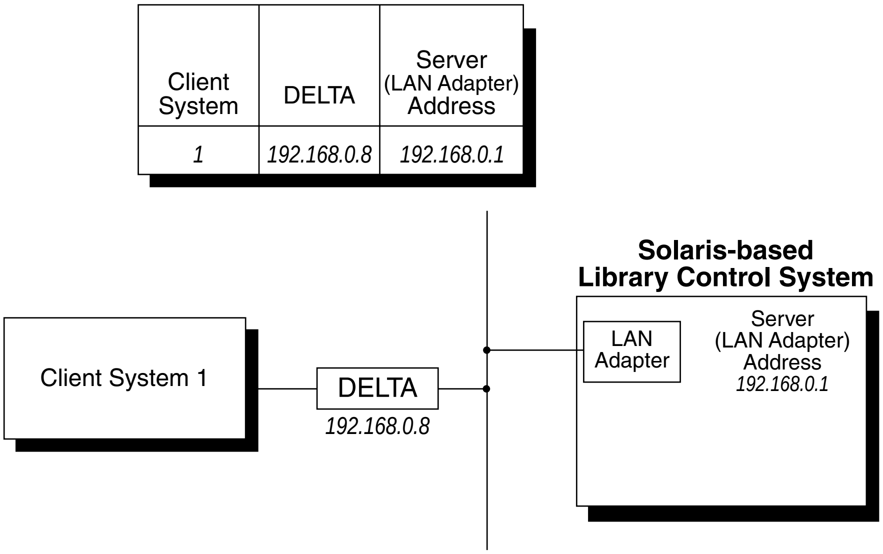
```
PROCESS,CSC    TYPE,TSAM  ;
PASSWORD,CSCPW  ;
INTERNET-ADR,IAETH1
```

*Relevant CSC configuration (SYS$LIB$*CSC-PARM.CSC$PARAM)*

```
INTERFACE CDIINT USES CDI
INTERFACE CMSINT USES CMSA,CSC,CSCPW
PATH GAMMA USES CMSINT CONNECTS 192,168,0,7 TO 192,168,0,1
PATH ALPHA USES CDIINT CONNECTS 192,168,0,3 TO 192,168,0,1
```

2.6.8 CPCOMM with one adapter / One LAN / One Server Address

Figure 2-8 shows the connection of one client to one server through CPCOMM.



Client System	DELTA	Server (LAN Adapter) Address
1	*192.168.0.8*	*192.168.0.1*

Figure 2-8. One Client/TSAM/One LAN

2.6.8.1 Configuration Statements

CPCOMM configuration for a TSAM process for CSC

```
    PROCESS,CSC   PASSWORD,CSCPW
```

*Relevant CSC configuration (SYS$LIB$*CSC-PARM.CSC$PARAM)*

```
    INTERFACE CPCOMMA USES CPCOMMA,CSC,CSCPW
    PATH DELTA USES CPCOMMA CONNECTS 192,168,0,8 TO 192,168,0,1
```

3. USING COMUS TO BUILD AND INSTALL CSC

This chapter describes how to register, build, and install the CSC product group. Examples of the COMUS dialogue provide detail for each step of the process.

Before performing any of the procedures in this chapter, you should have:

- reviewed the pre-installation requirements outlined in the section, “Before You Install the CSC Product Group” in Chapter 1.
- configured your communication method as described in Chapter 2, “Configuring the Communication Method.”

3.1 OVERVIEW

You begin the CSC product group build and installation process with the CSC product tape. The installation with no CSC updates consists of the following major steps:

- Perform a COMUS REGISTER of the CSC product tape.
- Perform a COMUS BUILD of the CSC product group, and then start the runstream created by the BUILD.
- Perform a COMUS INSTALL of the CSC product group using the Output Master Tape produced in Step 2.
- Verify the CSC configuration elements in the CSC parameter file and update if necessary.

If you are applying updated object modules, the installation consists of the following steps:

- If this is the initial installation of CSC, perform a COMUS REGISTER of the CSC product tape.
- Perform a CSC BUILD, and specify that you have an update at the appropriate prompt. Start the runstream created by the build.
- Perform a COMUS install using the Output Master Tape produced in Step 2.

NOTE

If you prefer to use the SOLAR method of installation, first complete the COMUS build process described in this chapter. Then, refer to Chapter 4, "Using SOLAR to Install CSC."

3.1.1 About CSC Interim Corrections

Updates to the CSC product are typically released in the form of an Interim Correction or IC. An IC is a file that is used during the CSC product build to update CSC. You specify the IC file in response to the prompt "`Do you have a CSC update/feature tape/file to apply? Y/<N>`."

If you have access to Unisys Web based support, the latest Interim Correction can be found in the Support Database in the PVP (Product Validation Profile) document for CSC. If you do not have access the Unisys Web based support, your Unisys Support Representative can obtain Interim Corrections for you.

3.1.2 About COMUS

In this chapter (and in this entire guide), we assume that you already know how to use COMUS, and so do not discuss your possible responses to the standard COMUS prompts. The following explanation of the CSC installation process is limited to those COMUS prompts specific to CSC. For more information on COMUS, please refer to the *COMUS Reference Manual.*

In general, remember that the default for each COMUS prompt is listed after the prompt in angle brackets (<>). If you press XMIT only, you accept the default for that prompt. You can get help on any prompt by entering a question mark (?). You can abort a COMUS procedure by entering \CANCEL in response to any prompt.

3.2 REGISTER THE CSC PRODUCT TAPE

The first step in the process of installing the CSC product group is to REGISTER the CSC product tape with COMUS. This describes CSC to COMUS by copying information from the product tape to the COMUS database. To REGISTER the CSC product tape with COMUS, you should follow these steps:

1. If your project ID differs from the COMUS database qualifier, enter the following command from a demand terminal:

   ```
   @QUAL database-qualifier
   ```

2. From COMUS, perform a REGISTER of the CSC product tape.

   To REGISTER the product tape by *reel-id*, where *reel-id* is the volume serial number (volser) of the CSC product tape, enter:

   ```
   REGISTER REEL=reel-id PRODUCT=CSC,4R1
   ```

   COMUS will display messages confirming the REGISTER, then return to the COMUS COMMAND? prompt.

3.3 COMUS BUILD OF CSC

The second step is to perform a COMUS BUILD of the CSC product group, using the COMUS ‘Q’ option with the BUILD command. Using BUILD,Q tells COMUS to redefine the local site parameters based on your responses. CSC will convert parameters from a previous install to 4R1 parameters. However, you can reset them to the release defaults by answering “Y” to the appropriate prompt.

You begin at the COMUS COMMAND prompt by entering:

```
BUILD,Q CSC,4R1
```

The COMUS BUILD is an interactive, prompt-driven process. Figure 3-1 shows the prompts you’ll encounter. Notice that some of the statements in the Figure 3-1 are set in bolded type. There are separate sections following the figure that explain each bolded statement in the COMUS BUILD.

```
COMMAND ?‣BUILD,Q CSC,4R1
Redefine the product defaults (LIST, Y, or <N>) ? ‣Y
Define CSC default values
For each default question you may enter one of the following responses:
  1. An appropriate value.
  2. A null string or spaces will maintain the current default value.
  3. QUERY - This keyword causes COMUS to ask for the default on every BUILD
     of the product.
  4. BLANK - This keyword sets the value of the default to null.
Default generation recovery mode (ON or <OFF>) ? ‣
Default project id (<CSC4R1>) ? ‣
Default run id (<CSCGEN>) ? ‣
Default run options (<A>) ? ‣
Default run priority (<A>) ? ‣
Default tape equipment type (<U47L>) ? ‣
Default tape assign options (<TF>) ? ‣
Default generation type (DISK/DISK or <TAPE/TAPE>) ? ‣
Permanent SGSs (<END>) ? ‣
CSC 3R5 parameters have been converted to 4R1
Reset the CSC Parameters to Release defaults ? Y/<N> : ‣
CSC defaults complete
Project id for this generation (<CSC4R1>) ? ‣
Run id for this generation (<CSCGEN>) ? ‣
MASTER - reel/file./<> ? ‣L10648
Generation id ? ‣CSC4R1
Generation heading (<>) ? ‣CSC 4R1
Generation reason (<END>) ? ‣CSC 4R1 build
Generation reason (<END>) ?
New change number (<END>) ? ‣
Do you have a CSC update/feature tape/file to apply? Y/<N> ‣
OMASTER - reel/file./<> ? > ‣BLANK,TF,U47L,365
Output destination: PRINTER,TAPE,MASTER,filename <P>/T/M/f
Begin type 2 SGS definition
Parameters affecting COMUS installed CSC components
     (A)- CANACT    <0>
     (B)- CANPRJ    <CSC-4R1>
     (C)- USERID    <CSC>
     (D)- CSCFLE    <SYS$LIB$*CSC>
     (E)- CSCPARM   <SYS$LIB$*CSC-PARM>
     (X)- No Change
Enter parameter selection or X for no change:  selection/<X> ‣
Do you want CDI configured?<Y>/N ‣
Input value for CDINETCB BDI - <0400410> : ‣
Do you want CSC User Interface?<Y>/N ‣
Parameters affecting the CSCUI exits
     (A)- CSCUICBA  <0400302>
     (B)- CSCUICBB  <0400303>
     (C)- CSCUILOG  <50302>
Enter parameter selection or X for no change: selection/<X> ‣
Do you want to configure for a C2 security environment? Y/<N> ‣Y
Parameters affecting C2 Security
     (A)- C2SEC  <Y>
     (B)- GATEBDI <0401206>
     (C)- GATECDI <0401203>
     (X)- No Change
Enter parameter selection of X for no change: selection/<X> ‣
Additional SGSs (<END>) ? ‣
The runstream has been saved in 'CSC*COMRUN(1).1/CSC4R1'
View the runstream (Y or <N>) ? ‣N
Print a copy of the runstream (Y or <N>) ? ‣Y
HDG? ‣CSC's copy of the 3-5-1 build
A copy of the runstream has been sent to PR
```

```
Start the runstream (Y or <N>) ? ‣N
The runstream has been saved in 'CSC*COMRUN(1).1/CSC4R1'
UPDATING ACCESS FILES ...
ACCESS FILES HAVE BEEN UPDATED
BUILD TASK COMPLETED  *************************
COMMAND ? ‣
```

Figure 3-1. CSC COMUS BUILD Prompts

Figure 3-1 shows only the top-level COMUS prompts. CSC-specific prompts are numbered and boldface. If your response to a prompt takes you to sub-level prompts (for example, “Enter communications mode”), those sub level prompts are discussed under the heading for the related top level prompt. Following is a discussion of each of these top level prompts and any sub level prompts, and of your possible responses.

3.3.1 MASTER - reel/file./<>

Enter the volser of the CSC product tape. If you are using a Tape Library Management System (TLMS), the CSC product tape is considered a “foreign” tape because it is not yet in your TLMS. Therefore, you must either assign the release tape with a bypass of the TLMS label processing, or enter the volser into the TLMS database.

3.3.2 Do you have a CSC update/feature tape/file to apply?Y/<N>

Answer Y if you are applying object module updates to CSC. You can apply the updates using a cataloged disk or tape file, or an update tape.

To use a cataloged disk or tape file, enter the filename at the first prompt and the file type at the second prompt. Or, enter the reel number of the update tape at the first prompt.

```
UPDTAPE - reel/file./NONE/<> ? ‣csc*update-4R1.
is CSC$UPDATE-4R1 a cataloged tape or disk file? D/<T> ‣d
```

or

```
UPDTAPE - reel/file./NONE/<> ? ‣112346,tf,u471
```

3.3.3 OMASTER - reel/file./<>

This prompt allows you to specify the parameters for the CSC Output Master Tape, which will hold the new runstream created by the COMUS BUILD. Your response should contain four parameters in the following syntax:

```
filename. | reel-id[,assign,equipment,expires]
```

where you enter:

- a *filename* or *reel-id* identifying the Output Master Tape. To assign a scratch cartridge tape, you can specify "BLANK" or null for this parameter. For example, you can use either the "BLANK,*assign,equipment,expires*" or the "*,assign,equipment,expires*" syntax to specify a scratch cartridge tape.
- an *assign* option, such as TF (for a labeled tape) or TJ (for an unlabeled tape).
- an *equipment* option, such as T (for round tapes), U47NL (for a standalone cartridge drive), or U47L (for a cartridge drive in the ACS).
- an *expires* option specifying the number of days until the cartridge tape expires, in days.

For example, you might enter:

```
L12345.,TJ,U47L,5
```

to assign the unlabeled cartridge tape "L12345", which expires in 5 days, on a cartridge drive in an ACS.

Alternatively, you might enter:

```
BLANK,TJ,U47L,5
```

to assign an unlabeled scratch cartridge tape, which expires in 5 days, on a cartridge drive in an ACS.

3.3.4 Type 2 SGS definition

If you're installing CSC for the first time (or if you are updating from a previous level of CSC and have different configuration parameters), you need to specify your current configuration using Type 2 SGS parameters.

```
Parameters affecting COMUS installed CSC components
   (A)- CANACT     <0>
   (B)- CANPRJ     <CSC-4R1>
   (C)- USERID     <CSC>
   (D)- CSCFLE     <SYS$LIB$*CSC>
   (E)- CSCPARM    <SYS$LIB$*CSC-PARM>
   (X)- No Change

Enter parameter selection or X for no change: selection/<X>
```

The default values for these parameters appear in angle brackets (<>). These defaults are based on the previous CSC generation (if any). You can update any or all of these parameters by entering the corresponding menu letter (A - E). You can enter X to accept the current values.

If you enter a letter from ‘A’ through ‘E’ to update a parameter, COMUS will prompt you to enter the new value for that parameter. Following are discussions of each of these parameters.

(A) CANACT Parameter

CSC uses this value as the account number on all generated runstreams for CSC. The account number must be one that won’t terminate based on the number of pages, time used, or the number of system log records written. The default is *0*.

(B) CANPRJ Parameter

CSC uses the CANPRJ parameter as the project id on all generated runstreams for CSC. The default is *CSC-4R1*.

(C) USERID Parameter

This value represents the user id for all generated runstreams for CSC. The default is *CSC*.

(D) CSCFLE Parameter

Enter the qualifier and filename where you want to install the product files. Any OS 2200 standard filename is valid. The default value for CSCFLE is *SYS$LIB$*CSC*.

(E) CSCPARM Parameter

Enter the qualifier and filename where you want all your CSC parameter elements to be saved. Any OS 2200 standard filename is valid. The default value for CSCPARM is *SYS$LIB$*CSC-PARM*.

3.3.5 CDI Interface

CDI is one of the TCP/IP communications method to connect to the server. The others are CMS and CPCOMM.

If you select N, CDI will not be configured. If you select Y, COMUS will continue with the following BDI request sub-level prompt:

```
   Input value for CDINETCB BDI - <0400410> : ‣
```

3.3.6 CSC User Interface

The CSC User Interface (CSCUI) lets programs receive notifications from CSC processing and submit requests to CSC. These functions allow a TLMS to interface with CSC. For more information on CSCUI, refer to the *CSCUI Programmer's Reference Manual.*

If you select Y to enable CSCUI, COMUS will continue with the BDI request sub-level prompt. If you enter N, the BUILD process is complete.

Configuring CSCUI Parameters

If you select Y to enable CSCUI, COMUS displays the following sub-level prompts:

```
Parameters affecting the CSCUI exits

   (A)- CSCUICBA  <0400302>
   (B)- CSCUICBB  <0400303>
   (C)- CSCUILOG  <50302>

Enter parameter selection or X for no change:  selection/<X>
```

This dialogue lets you configure CSCUI parameters. The first two parameters are CSCUI BDIs. The third parameter is the log record number that will identify all CSCUI events in the system log file. You can change the value for any parameter listed by entering the letter for that exit (A - C), then inputting the new value. Enter X to accept the default values and end CSCUI parameter configuration.

3.3.7 C2 Security

The C2 Security prompts are for sites that intend on running CSC in a C2 or B1 security environment that have the Common Bank Protection feature enabled. This environment requires that all common banks have gates defined for each bank that may be called.

(A) C2SEC Parameter

This value determines if C2 Security is enabled. A value of "Y" indicates that CSC will be built for a C2 environment. A value of "N" will disable the C2 Security support, and will generate a CSC for a non-C2 security environment.

(B) GATEBDI Parameter

This value is the BDI that will be used as the gate bank for the CSCUICBA common bank.

(C) GATECDI Parameter

This value is the BDI that will be used as the gate common bank for the CDINETCB common bank.

3.4 START THE COMUS/CSC BUILD RUNSTREAM

Once you've responded to the COMUS prompts, COMUS creates and saves the COMUS/CSC BUILD runstream on the CSC Output Master Tape and then returns you to the COMUS COMMAND?prompt. At this point, you can start the BUILD runstream through COMUS by answering Y to the “Start the runstream?” prompt. Alternatively, you can exit COMUS and start the COMUS/CSC BUILD runstream by entering the following from a demand terminal:

```
@START runstream
```

where *runstream* is the name of the COMUS/CSC BUILD runstream you just created.

3.5 COMUS INSTALL OF CSC

The third step in the process of registering, building, and installing the CSC product group is to perform a COMUS INSTALL of CSC. You begin from the COMUS COMMAND?prompt by entering:

```
INSTALL CSC,4R1
```

The COMUS INSTALL, like the COMUS BUILD, is an interactive, prompt-driven process. Figure 3-2 shows the prompts you’ll encounter. In Figure 3-2, the CSC-specific prompts are bolded and numbered. Following is a discussion of each of these prompts and of your possible responses.

```
   COMMAND ? ▸INSTALL CSC,4R1
❶  ENTER REEL,<REELNUM> OR FILE,<FILENAME> ? ▸L12345
   Enter any permanent SGS's for product CSC 4R1
   SGS or <END> ? ▸
   Enter the project id for this installation (<>) ? ▸
   Enter reason for installation of CSC 4R1 ? ▸
   Reason or <END> ? ▸
   Enter any additional SGS's for CSC 4R1
   SGS or <END> ? ▸
   The runstream has been saved in '*COMRUN.INS-1'
   View the runstream (Y or <N>) ? ▸
   Print a copy of the runstream (Y or <N>) ? ▸
   Start the runstream (<Y> or N)? ▸
   The runstream has been saved in  '*COMRUN.INS-1'
   INSTALL TASK COMPLETED  ************************
   COMMAND ? ▸
```

Figure 3-2. CSC COMUS INSTALL Prompts

3.5.1 Enter reel or file

Specify the location of the CSC Output Master Tape, which you created at the end of the COMUS BUILD process, either by volser or filename. After the installation runstream is started and the process is complete, refer to Chapter 5, “Completing the Installation Process.”

4. USING SOLAR TO INSTALL CSC

This chapter describes how to use SOLAR to register and install CSC, after your COMUS build of the product group is complete. Examples of the SOLAR dialogue provide detail for each step of the process.

Before performing the SOLAR installation in this chapter, you should have:

- reviewed the pre-installation requirements outlined in the section, “Before You Install the CSC Product Group,” in Chapter 1.

- configure your communication method as described in Chapter 2, “Configuring the Communication Method.”

- performed a COMUS build of the CSC product group as outlined in Chapter 3, “Using COMUS to Build and Install CSC.”

4.1 OVERVIEW

You begin the CSC installation process with the CSC Output Master Tape (OMASTER) created during the COMUS build in Chapter 3. The SOLAR installation consists of the following major steps:

- Call the SOLAR processor.
- Register the CSC software package.
- Install the registered package.
- Alternatively, CSC can be installed using the SOLAR INSTALLPKG run. This method is described at the end of this chapter.

NOTE

If you prefer to use the COMUS method of installation, refer to Chapter 3, "Using COMUS to Build and Install CSC."

4.1.1 About SOLAR

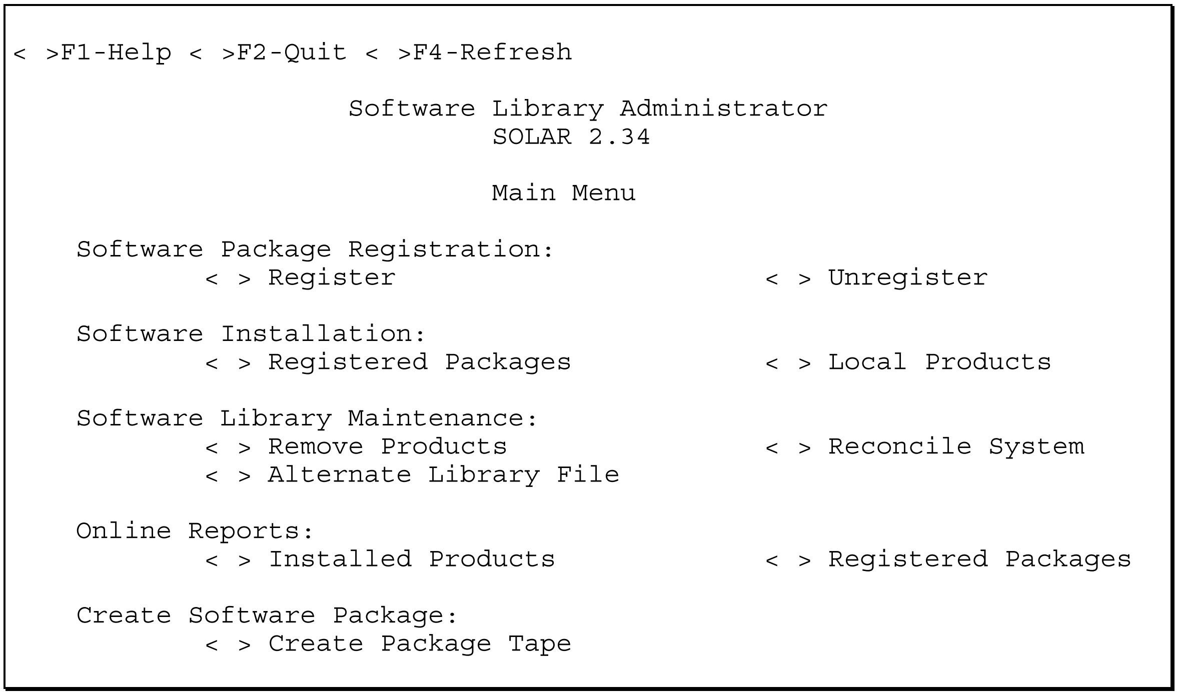
In this chapter (and in this entire guide), we assume that you already know how to use SOLAR, therefore, we do not illustrate every available screen, nor do we describe every available utility. Instead, this chapter provides the basic information required to register and install CSC.

4.2 CALL THE SOLAR PROCESSOR

The first step in the process of installing CSC is to call the SOLAR processor by entering the following command from a demand terminal:

```
@SOLAR
```

When started, SOLAR displays a main menu similar to the one on the following page.

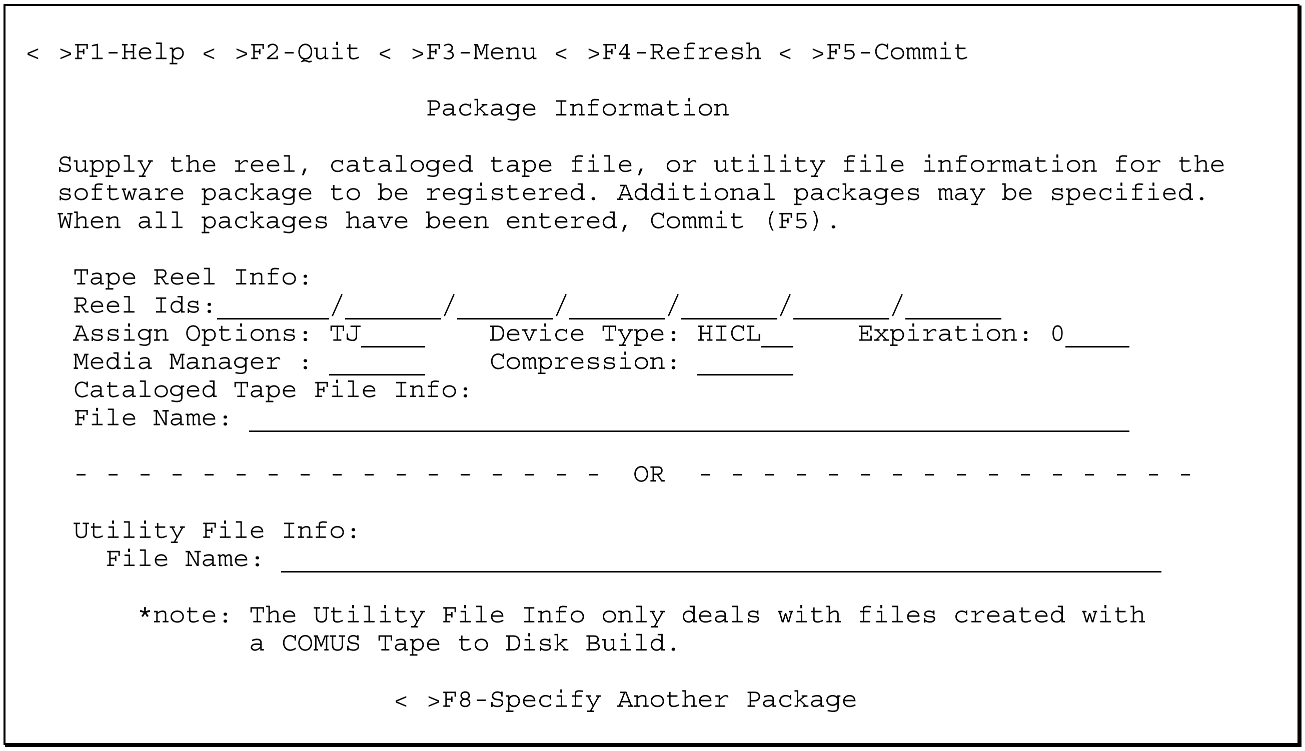


```
< >F1-Help < >F2-Quit < >F4-Refresh

                     Software Library Administrator
                              SOLAR 2.34

                              Main Menu

    Software Package Registration:
            < > Register                      < > Unregister

    Software Installation:
            < > Registered Packages           < > Local Products

    Software Library Maintenance:
            < > Remove Products               < > Reconcile System
            < > Alternate Library File

    Online Reports:
            < > Installed Products            < > Registered Packages

    Create Software Package:
            < > Create Package Tape
```

Figure 4-1. SOLAR Main Menu

4.3 REGISTER THE CSC SOFTWARE PACKAGE

Once SOLAR is started and the main menu is displayed, you can register your Output Master Tape by positioning the cursor between the angle brackets (< >) labeled “Register,” under the “Software Package Registration” function, and transmitting. When complete, SOLAR starts a utility called PKGREG and displays a package information screen similar to the one in Figure 4-2.



```
< >F1-Help < >F2-Quit < >F3-Menu < >F4-Refresh < >F5-Commit

                         Package Information

  Supply the reel, cataloged tape file, or utility file information for the
  software package to be registered. Additional packages may be specified.
  When all packages have been entered, Commit (F5).

   Tape Reel Info:
   Reel Ids:_______/______/______/______/______/______/______
   Assign Options: TJ____    Device Type: HICL__    Expiration: 0____
   Media Manager : ______    Compression: ______
   Cataloged Tape File Info:
   File Name: _______________________________________________________

   - - - - - - - - - - - - - - - - -  OR  - - - - - - - - - - - - - - - -

   Utility File Info:
     File Name: _______________________________________________________

       *note: The Utility File Info only deals with files created with
              a COMUS Tape to Disk Build.

                       < >F8-Specify Another Package
```

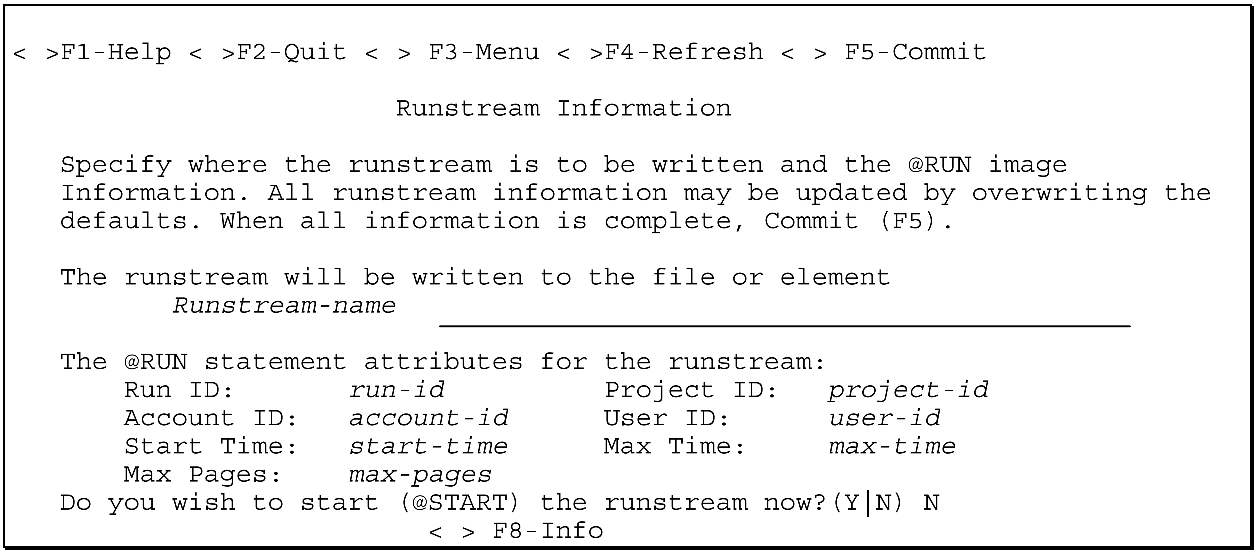
Figure 4-2. Package Information Screen

Table 4-1 describes the fields used on the package information screen.

Table 4-1. Fields Used on the Package Information Screen

Field	Description
Reel-id	The reel-id of the CSC Output Master Tape (OMASTER) created during the COMUS build.
Options	The tape assign option(s) for the specified reel-id. Use TF for labeled tapes and TJ for unlabeled tapes.
Type	The equipment type for the specified reel-id. If the tape resides in the ACS, use U47L. Otherwise, use U47.
Expiration	The number of days a logical file written to a labeled package tape is write-protected. This field only applies to labeled tapes.
Mmgr	The media manager specification for the specified reel-id.
Compression	The data compression used for the specified reel-id.
Cataloged-tape-filename	The filename of the cataloged tape file used to assign the tape.
Utility-tape-filename	The name of a utility file created during a COMUS build.

After you provide the necessary information and commit the data (F5), SOLAR displays a runstream information screen similar to the one in Figure 4-3.



```
< >F1-Help < >F2-Quit < > F3-Menu < >F4-Refresh < > F5-Commit

                        Runstream Information

   Specify where the runstream is to be written and the @RUN image
   Information. All runstream information may be updated by overwriting the
   defaults. When all information is complete, Commit (F5).

   The runstream will be written to the file or element
          Runstream-name   ___________________________________________

   The @RUN statement attributes for the runstream:
       Run ID:       run-id          Project ID:   project-id
       Account ID:   account-id      User ID:      user-id
       Start Time:   start-time      Max Time:     max-time
       Max Pages:    max-pages
   Do you wish to start (@START) the runstream now?(Y|N) N
                          < > F8-Info
```

Figure 4-3. Runstream Information Screen

Table 4-2. Fields Used on the Runstream Information Screen

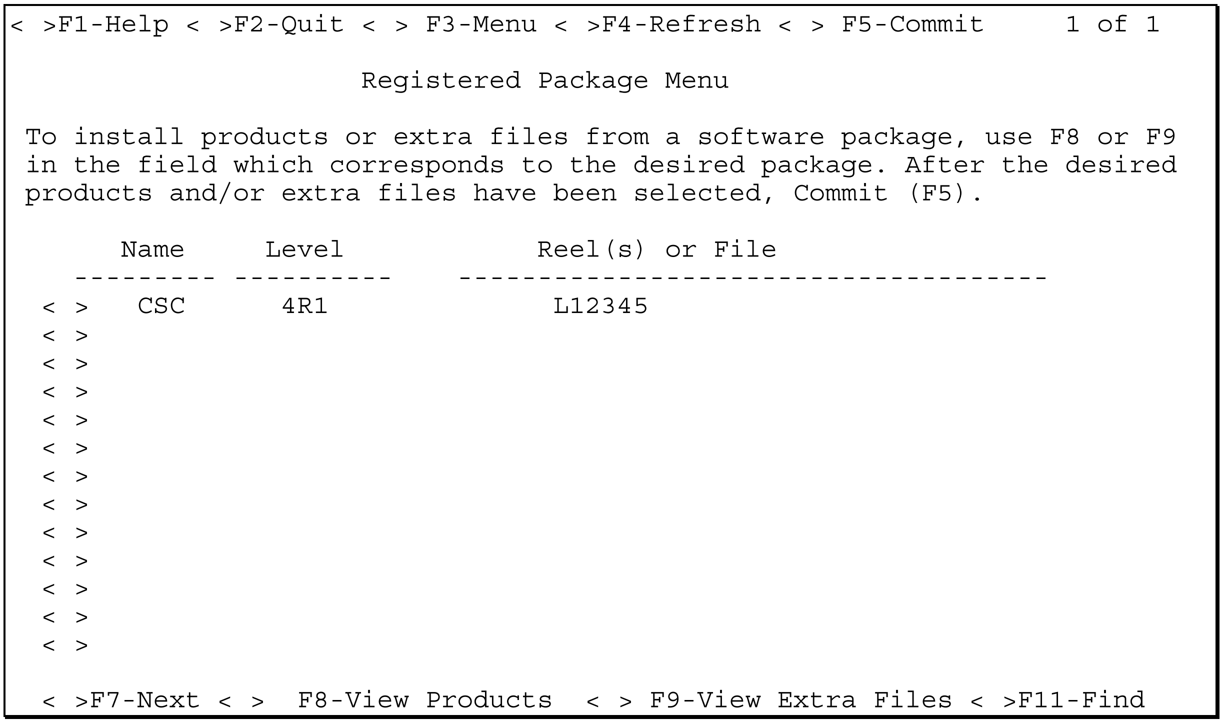
Field	Description
Runstream-name	The name of the file or element where SOLAR writes the runstream.
Run-id	The run-id of the generated runstream.
Project-id	The project-id of the generated runstream.
Account-id	The account number for the @RUN statement in the runstream that SOLAR creates.
User-id	The user-id for the @RUN statement in the runstream that SOLAR creates.
Start-time	The start time that you want specified on the @RUN statement in the runstream that SOLAR creates.
Max-time	The maximum time that you want specified on the @RUN statement in the runstream that SOLAR creates.
Max-pages	The maximum pages that you want specified on the @RUN statement in the runstream that SOLAR creates.

In addition to supplying the information described in Table 4-2, you can optionally start the registration runstream by specifying “Y” after the question, "Do you wish to start (@START) the runstream now?" If you answer with "N," you must start the runstream manually, using a demand terminal or the system console.

After you provide the necessary information and commit the data (F5), SOLAR redisplays the main menu.

4.4 INSTALL THE REGISTERED PACKAGE

Once the main menu is redisplayed, you can install your registered package by positioning the cursor between the angle brackets (< >) labeled "Registered Packages," under the "Software Installation" function, and transmitting. When complete, SOLAR starts a utility called PRODLD and displays a registered package menu similar to the one in Figure 4-4.



```
< >F1-Help < >F2-Quit < > F3-Menu < >F4-Refresh < > F5-Commit     1 of 1

                      Registered Package Menu

 To install products or extra files from a software package, use F8 or F9
 in the field which corresponds to the desired package. After the desired
 products and/or extra files have been selected, Commit (F5).

       Name     Level            Reel(s) or File
    --------- ----------    -------------------------------------
  < >   CSC      4R1              L12345
  < >
  < >
  < >
  < >
  < >
  < >
  < >
  < >
  < >
  < >
  < >
  < >

  < >F7-Next < >  F8-View Products  < > F9-View Extra Files < >F11-Find
```

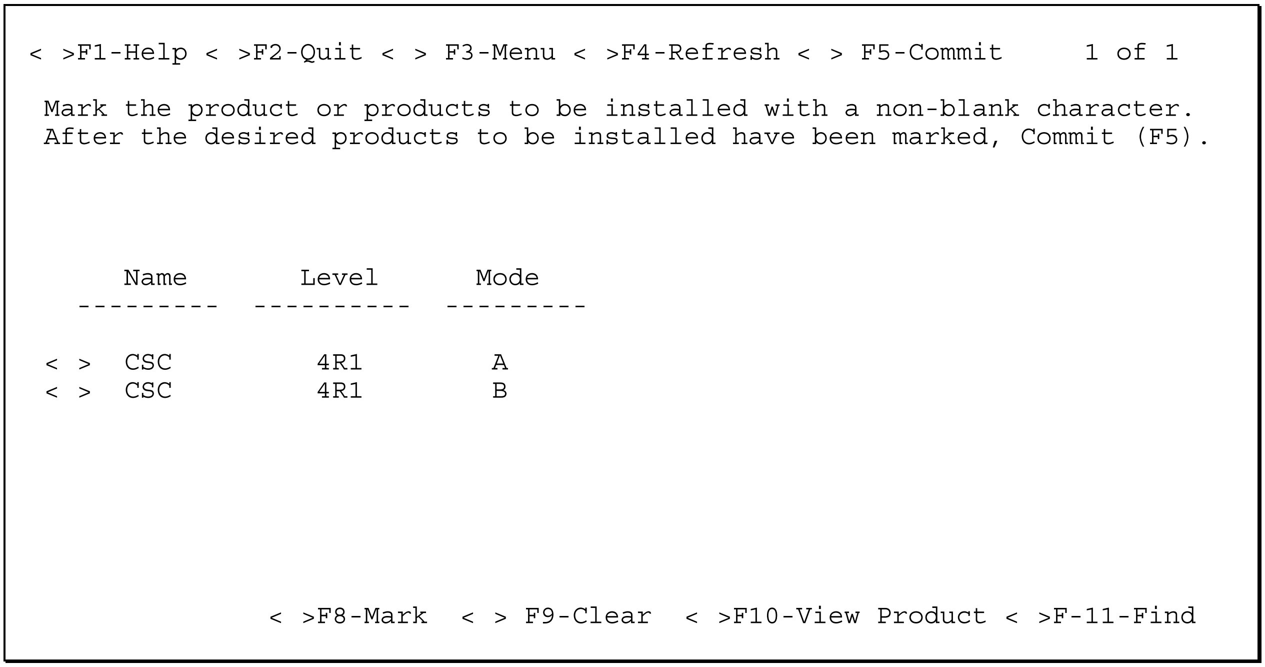
Figure 4-4. Registered Package Menu

Table 4-3 describes the fields used on the registered package menu.

Table 4-3. Fields Used on the Registered Package Menu

Field	Description
Product-name	The name of the software product registered with SOLAR.
Level	The release level of the registered product.
Reel-or-file	The reel-id or filename that was specified during the SOLAR registration.

To create an installation runstream, position the cursor between the angle brackets (< >) that appear alongside the software package you wish to install, and press F8 to “View Products.” When complete, SOLAR displays a product selection screen similar to the one in Figure 4-5.



```
< >F1-Help < >F2-Quit < > F3-Menu < >F4-Refresh < > F5-Commit     1 of 1

 Mark the product or products to be installed with a non-blank character.
 After the desired products to be installed have been marked, Commit (F5).




      Name       Level      Mode
   ---------  ----------  ---------

 < >  CSC         4R1        A
 < >  CSC         4R1        B






               < >F8-Mark  < > F9-Clear  < >F10-View Product < >F-11-Find
```

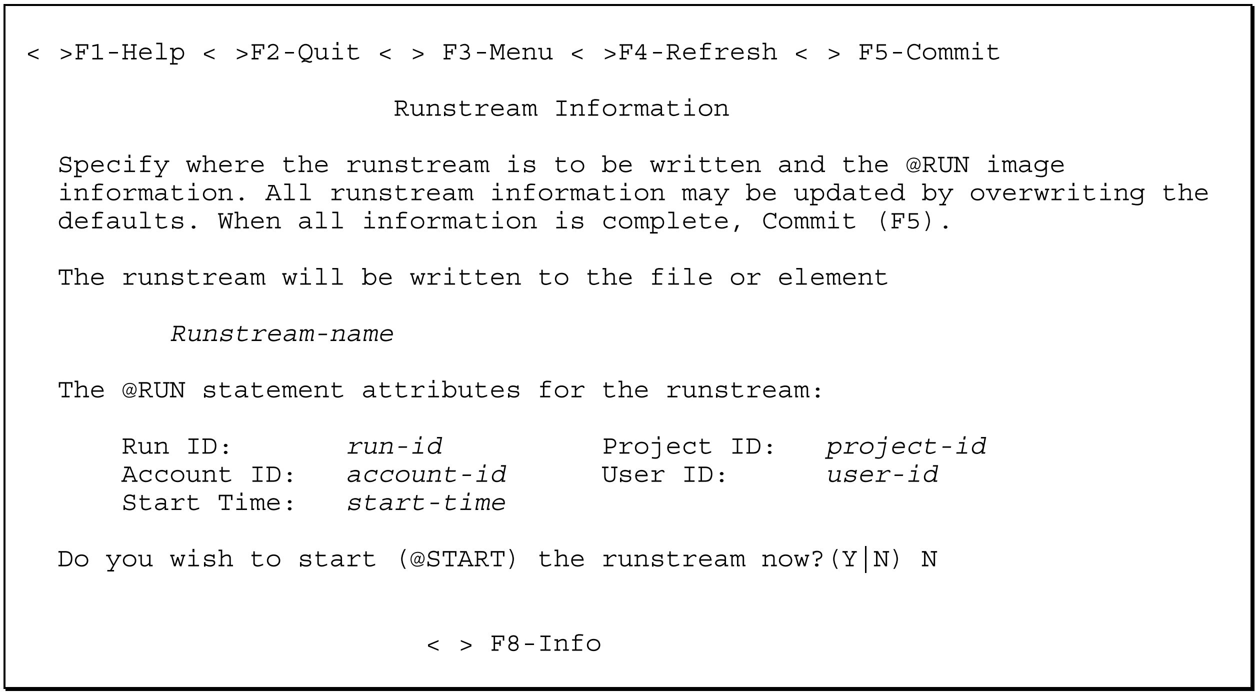
Figure 4-5. Product Selection Screen

CSC Installation Modes

CSC has a two installation modes. The default installation mode is mode A. The Mode B install is identical to the Mode A install except that it does not register any of the alternate file processors and it does not copy the runstreams to SYS$LIB$*RUN$.

Once again, position the cursor between the angle brackets (< >) that appear alongside the software package you selected, and press F8 to “mark” the entry.

When finished, commit your selection (F5). SOLAR will then display a runstream information screen similar to the one in Figure 4-6.



```
< >F1-Help < >F2-Quit < > F3-Menu < >F4-Refresh < > F5-Commit

                       Runstream Information

  Specify where the runstream is to be written and the @RUN image
  information. All runstream information may be updated by overwriting the
  defaults. When all information is complete, Commit (F5).

  The runstream will be written to the file or element

         Runstream-name

  The @RUN statement attributes for the runstream:

      Run ID:        run-id          Project ID:   project-id
      Account ID:    account-id      User ID:      user-id
      Start Time:    start-time

  Do you wish to start (@START) the runstream now?(Y|N) N


                         < > F8-Info
```

Figure 4-6. Runstream Information Screen

Refer to Table 4-2 for descriptions of the fields used on the runstream information screen.

In addition to supplying the information described in Table 4-2, you can optionally start the installation runstream by specifying “Y” after the question, “Do you wish to start (@START) the runstream now?” If you answer with “N,” you must start the runstream manually using a demand terminal or the system console.

After you provide the necessary information and commit the data (F5), SOLAR will redisplay the main menu.

After the installation runstream is started and the process is complete, refer to Chapter 5, “Completing the Installation Process.” (Please see the following note.)

NOTE

It's important that you check the output listing created by the SOLAR PRODLD runstream to verify that the installation was successful. If SOLAR encounters an error, PRODLD will fin, but give no external indication of the error. Also, SOLAR does not deinstall product files prior to the installation as COMUS does. Any files that existed before the SOLAR installation was started and subsequently failed, will still be present after the installation fins.

4.5 USING INSTALLPKG TO INSTALL CSC

CSC can also be installed using the SOLAR INSTALLPKG run. It registers and installs the tape using a batch job started at the system console. The INSTALLPKG session and prompts follow:

```
‣ ST INSTALLPKG,,,0/INSTALL

      INSPKG START

0-INSPKG*INSTALL FROM <TAPE>, DISK, OR RSS

‣ 0

0-INSPKG*ENTER REEL NUMBER

‣ 0 L12345,TF,U47L

0-INSPKG*INSTALL DEFAULT MODES? <Y>/N

‣ 0

      INSPKG FIN
```

When this run FINs, the CSC product file and parameter file have been created and CSC is installed.

5. COMPLETING THE INSTALLATION PROCESS

This chapter describes the few remaining steps that must be considered before the CSC installation is complete.

5.1 OVERVIEW

To complete the CSC installation process, you must consider the following steps:

1. Verify CSC parameter elements.
2. Configure Demand Terminal Security.
3. Configure CSCUI.
4. Configure the Dynamic Timer feature.
5. Copy updated configuration elements to SYS$LIB$*CSC-PARM.
6. Configure NODE statements for input to the EXEC generation.
7. Review SERVER CONCERNS.

5.2 CSC PARAMETER ELEMENTS

The build and install process will upgrade and/or restore your existing configuration elements. The first step is to verify the elements within CSC. All of the following elements can be found in the CSC parameter file as specified in the CSCPARM parameter. The default filename is SYS$LIB$*CSC-PARM. To configure these elements, you can use the @ED processor or any other editor to create ASCII elements.

The parameter elements are:

- CSC$PARAM
- CSC$MESSAGE
- CSC$DRIVE
- NCS$PARAM

5.2.1 CSC$PARAM

CSC$PARAM controls the CSC runtime environment. Use a text editor to enter configuration directives into the CSC$PARAM element. The general syntax for a configuration directive is:

```
directive [=] parm-1, parm-2, ..., parm-n
```

where *directive* is one of the following:

AUTOMOUNT
CDIBDI
CSCUIBDI
DEFAULT_LABELED_POOL

DEFAULT_UNLABELED_POOL
EJECT_3R4
EJECT_RETRY
ERINDEX
INTERFACE
KEYIN
PATH
SERVER_TYPE
SIGNON
TRANSLATE_POOL
UNDEFINED_POOL
USERID

When the equal sign is present, spaces before and after it are ignored.

These directives and their parameters are described below. You must enter each directive into CSC$PARAM. By OS 2200 convention, you must precede octal values by a 0 (zero). Figure 5-1 shows a sample CSC configuration.

```
CDIBDI=0400410,0141000
CSCUIBDI=0400302,0400303
ERINDEX=0275
DEFAULT_LABELED_POOL=5
DEFAULT_UNLABELED_POOL=6
INTERFACE CDIINT USES CDI
PATH ALPHA USES CDIINT CONNECTS 192,168,0,3 TO 192,168,0,1
```

Figure 5-1. Sample CSC Configuration Parameters

5.2.1.1 AUTOMOUNT

This configuration directive clears the automated mount switch, which disables the automated mounting of tape. This capability applies only to mount requests from OS 2200. BEFORE-MOUNT notifications are produced regardless of the setting of the switch. The syntax for AUTOMOUNT is:

```
AUTOMOUNT [=] state
```

Where *state* indicates if automated mounts will be performed. It must be "OFF" to disable CSC's automated mounting of volumes.

NOTE

When the automated mounting of volumes is disabled, it is the responsibility of the Tape Management System to register for BEFORE-MOUNT notifications and issue MOUNT requests.

5.2.1.2 CDIBDI

This configuration directive specifies the BDI number for CDI. The syntax for CDIBDI is:

```
CDIBDI [=] parm-1,0141000
```

where *parm-1* is the BDI number for CDI. The value of this parameter must match the number specified for CDINETCB during the COMUS BUILD. You *must* specify the reserved number 0141000 as *parm-2* in this directive.

NOTE

The CDIBDI parameter should not be specified without an INTERFACE parameter for CDI.

5.2.1.3 CSCUIBDI

This configuration directive specifies the BDI number for CSCUI. The syntax for CSCUIBDI is:

```
CSCUIBDI [=] parm-1,parm-2
```

where *parm-1* is the first BDI for CSCUI and *parm-2* is the second BDI for CSCUI. The value of *parm-1* must match the number specified for CSCUICBA during the COMUS BUILD. The value of *parm-2* must match the number specified for CSCUICBB during the COMUS BUILD.

5.2.1.4 DEFAULT_LABELED_POOL

This configuration directive specifies the default scratch pool number for labeled cartridge tapes. You must set up scratch pools when you configure the server. The syntax for DEFAULT_LABELED_POOL is:

```
DEFAULT_LABELED_POOL [=] parm-1
```

where *parm-1* is the default scratch pool number for labeled cartridge tapes. You should check with your system administrator for a valid pool number.

5.2.1.5 DEFAULT_UNLABELED_POOL

This configuration directive specifies the default scratch pool number for unlabeled cartridge tapes. You must set up scratch pools when you configure the server. The syntax for DEFAULT_UNLABELED_POOL is:

```
DEFAULT_UNLABELED_POOL [=] parm-1
```

where *parm-1* is the default scratch pool number for unlabeled cartridge tapes. You should check with your system administrator for a valid pool number.

5.2.1.6 EJECT_3R4

This configuration directive can disable the CSCUI EJECT enhancements introduced in CSC 3R4. The syntax is:

```
EJECT_3R4 [=] OFF
```

With this directive, programs that use the CSCUI EJECT request will behave exactly as they did prior to CSC 3R4.

5.2.1.7 EJECT_RETRY

This configuration directive specifies the delay between retries of CSCUI EJECT requests. It is applicable only if the EJECT_3R4 OFF directive is not present. The syntax for EJECT_RETRY is:

```
EJECT_RETRY [=] parm
```

where `parm` is the time in seconds between re-submissions of queued EJECT requests. Valid values for `parm` are from 20 to 600 inclusive. If this directive is not present, the default of 40 seconds is used.

5.2.1.8 ERINDEX

This configuration directive specifies the user ER index used by CSC. The syntax for ERINDEX is:

```
ERINDEX [=] parm-1
```

where *parm-1* is the ER index used by CSC. You *must* specify 0275 for this parameter.

5.2.1.9 INTERFACE

This configuration directive defines the interfaces to be used to communicate with the server. The syntax for INTERFACE is:

```
INTERFACE intf-name USES transport-type[,TI-information]
[STATUS,stat]
```

Intf-name is the identifier that CSC will use for this interface in operator commands and in program output. This name must be unique.

Transport-type tells what external software controls the paths for this transport interface. There can be only one INTERFACE statement for each transport type. Following are the transport types supported by CSC.

CDI
CMSA — CMS 1100 mode A installation

CMSTEST — CMS 1100 mode TEST installation
CPCOMMA — CPCOMM mode A installation
CPCOMMB — CPCOMM mode B installation
CPCOMMC — CPCOMM mode C installation
CPCOMMD — CPCOMM mode D installation

TI-information represents parameters that are specific to a type of transport interface. For CDI this is the BDI and entry address for calling CDI. For CMS or CPComm, this is the process name and password.

Stat is the initial state of this transport interface. It is either UP or DOWN.

Example

```
INTERFACE INTFA USES CDI,0400410,0141000
INTERFACE INTFB USES CMSA,PROCESSA,NOPSWD
INTERFACE INTFC USES CMSTEST,PROCESSA,SOMEPSWD STATUS,DOWN
```

5.2.1.10 IPADDR

In previous CSC levels, this configuration parameter specified the IP address of the server. This functionality is now achieved with the PATH parameter. The IPADDR parameter is no longer needed. If present, it is ignored.

5.2.1.11 KEYIN

This configuration directive allows the site to change the console keyin used by CSC. The syntax for KEYIN is:

```
KEYIN [=] keyin
```

where *keyin* is the keyin to be used by CSC.

5.2.1.12 PATH

This configuration directive defines the PATHs each interface can use. There must be at least one path per interface. The syntax for PATH is:

```
PATH path-name USES intf-name CONNECTS ca-address TO server-
address STATUS,stat
```

Path-name is the identifier that CSC will use for this path in operator commands and in program output. All path names must be unique.

intf-name is the name of the interface that this path will use.

Ca-address is the IP address of the communication adapter used by this path.

Server-address is the IP address on the server end of this communication path.

Stat is the initial state of this path. It is either UP or DOWN.

Example

```
PATH ALPHA USES INTFA CONNECTS 192,168,25,1 TO 192,168,25,50
   STATUS,DOWN
PATH GAMMA USES INTFB CONNECTS 192,168,25,4 TO 192,168,25,50
PATH DELTA USES INTFC CONNECTS 192,168,25,4 TO 192,168,25,50
PATH BRAVO USES INTFA CONNECTS 192,168,25,2 TO 192,168,25,50
PATH ECHO USES INTFA CONNECTS 192,168,25,2 TO 192,168,25,50
```

5.2.1.13 SERVER_TYPE

This configuration directive tells the type of server with which CSC will communicate. CSC internal timing and error handling are adjusted based on the server type. The syntax for SERVER_TYPE is:

```
SERVE_TYPE [=] type
```

where *type* is either ACSLS or NCS. The default is ACSLS.

5.2.1.14 SIGNON

This configuration directive defines the login information, which is required before the volrpt utility can be run on the Solaris-based server. The login information is site specific and may not be available to the DO-VOLRPT requester. The user name and password for logging on to the server are therefore included in the CSC configuration file. The syntax for SIGNON is:

```
SIGNON [=] user_name password
```

Where *`user_name`* is the login name for the Solaris-based server, and *`password`* is the password for user_name on the Solaris-based server. The user_name is generally *acsss*.

5.2.1.15 TRANSLATE_POOL

This configuration directive associates CTL-pool names with additional scratch pools that are defined on the server. The syntax for TRANSLATE_POOL is:

```
TRANSLATE_POOL CTL-pool,labeltype INTO ACS-pool
```

where *CTL-pool* is the CTL pool name to be used on the @ASG or @CAT statement; *labeltype* is the LABELED or UNLABELED keyword; and *ACS-pool* is the ACS pool number.

5.2.1.16 UNDEFINED_POOL

This configuration directive indicates what action to take when the CTL-pool name and label type cannot be translated to an ACS pool when processing a mount or scratch request. A DEFAULT value in the UNDEFINED_POOL MOUNT directive causes CSC to use the appropriate default pool to satisfy the mount request, and a REJECT value causes rejection of the request. A SAME-POOL value in the UNDEFINED_POOL SCRATCH directive causes CSC to scratch the tape to the pool with which the tape is currently associated, and a REJECT value causes rejection of the request. The syntax for UNDEFINED_POOL is:

```
UNDEFINED_POOL MOUNT ACTION IS action
```

where *action* is either the REJECT or DEFAULT keyword,

or:

```
UNDEFINED_POOL SCRATCH ACTION IS action
```

where *action* is either the REJECT or SAME-POOL keyword.

The default for the UNDEFINED_POOL MOUNT ACTION directive is REJECT. The default for the UNDEFINED_POOL SCRATCH ACTION directive is SAME-POOL.

5.2.1.17 USERID

This configuration directive specifies the userid to be used for the lock drive requests. This parameter can be up to 64 characters. The syntax for USERID is:

```
USERID [=] userid
```

5.2.2 CSC$MESSAGE

CSC$MESSAGE contains all messages displayed by CSC, with the exception of release level and internal number messages. In Figure 5-2, a portion of the CSC$MESSAGE element is shown.

```
U00000 %s initialization complete
R00001 DRIVE %s IS NOT AN ACS UNIT
R00002 VOLUME %s IS IN USE
R00003 VOLUME %s IS NOT IN THE ACS
R00004 VOLUME %s IS MISPLACED
R00005 VOLUME %s HAS AN UNREADABLE LABEL
I00006 %s DETECTED ANOTHER ACTIVE CSC
U00006 message 7
U00008 message 8
```

```
U00009 message 9
R00010 MOUNT %s ON %s COMPLETE
```

Figure 5-2. CSC$MESSAGE Element

Internally, CSC uses an index number which points to an entry in this message element and extracts the text for the desired message. The substitution directive, “%s”, is for variable data.

The alphanumeric number appearing at the beginning of each message is for search and reference purposes. *Do not change or delete these reference numbers.* The “U” prefix denotes undefined messages, “R” denotes references, and “I” is used for initialization and termination messages.

Although not recommended, you can alter these messages as long as the total number of substitution directives in each message stays the same. Otherwise, the resulting display will not contain full and complete information. You should not attempt to add or delete messages from this element. Otherwise, errors may occur.

5.2.3 CSC$DRIVE

CSC$DRIVE contains the cartridge drive mapping table for CSC. It must be configured on the client to match your specific cartridge drive configuration. The general syntax for each entry in the cartridge drive mapping table is:

`DRIVE` *`xxxxxx=acs#,lsm#,panel#,drive#`*

where *xxxxxx* is the logical cartridge drive name (the same name shown by an FS,TAPES display); *acs#* is the ACS number; *lsm#* is the LSM number; *panel#* is the number of the panel that contains the cartridge drive; and *drive#* is the cartridge drive number, counting down from the top of the panel.

Figure 5-3 shows an example CSC$DRIVE element.

```
   Drive ATAP0=0,0,9,0
   Drive ATAP1=0,0,9,1
   Drive ATAP2=0,0,9,2
   Drive ATAP3=0,0,9,3
   Drive ATAP4=0,0,10,0
   Drive ATAP5=0,0,10,1
   Drive ATAP6=0,0,10,2
   Drive ATAP7=0,0,10,3
   Drive ATAP8=0,0,2,0
   Drive ATAP9=0,0,2,1
   Drive ATAP10=0,0,2,2
   Drive ATAP11=0,0,2,3
```

Figure 5-3. CSC$DRIVE Element

5.2.4 NCS$PARAM

NCS$PARAM contains parameters that are unique to the NCS server. These parameters define site dependent responses for creating the terminal sessions used by the volume report and by the console command keyin. The `CFG_COMMAND` and `CFG_VOLRPT` parameters have identical formats. A generic description of their use is given here. Details specific to each command are presented in subsequent sections.

The CSC installation copies the following three files to the CSC parameter file:

1. NCS$PARAM/REQUIRED contains the statements that must be in NCS$PARAM for proper operation.

2. NCS$PARAM/DEFAULT tells the statements that are built into CSC.

3. NCS$PARAM/OPTIONAL contains statements that add possibly desirable changes to the CSC to NCS interface.

Following is the general flow of events for a volume report or a console command session.

1. CSC opens a connection to the server.
2. The server prompt for login information and CSC responds. This prompt and response continues until CSC establishes a terminal session with the server.
3. CSC waits for a prompt indicating that it is time to request the desired function. CSC then makes the request.
4. In the case of a volume report, CSC waits for a prompt that tells that the volume report is completed and then sends a request to retrieve the volume report output.
5. The information transfer phase is then entered. For console commands, CSC operator input is sent to the server and server responses are displayed on the operator console. For volume reports, the volume report output is received and written to the user specified file.
6. The console command session should exist for the duration of the CSC execution. If CSC detects a failure in this session, it waits for a short while and starts again at step 1, opening a connection.
7. For a volume report, CSC looks for a sentinel that the volume report is completed. When this arrives, CSC requests the termination of the terminal session. The user requested volume report completes when the session is ended.

Within CSC, volume reports and console commands use state based algorithms. The internal state processing can be controlled using the `CFG_COMMAND` and `CFG_VOLRPT` CSC parameters. Within each processing state, CSC waits for site specific character strings to arrive from the server. These are prompts for CSC to perform one or more of the following actions:

- Send a response to the server
- Display a message on the console
- Change to another processing state

The server prompts, responses, console messages, and state transitions can all be defined using the `CFG_COMMAND` and `CFG_VOLRPT` parameters. The following table shows the relation between CSC session states and the general flow listed above.

State	**Reference**	**Description**
LOGIN	1, 2	Establish a terminal session with the server.
SUBMIT	3	Submit a request to enter console mode or to execute the volume report utility.
RETRIEVE	4	Submit a request to retrieve the output of the volume report utility.
CONSOLE or TRANSFER	5	Transfer volume report or console command data.
FINISH	7	Terminate the terminal session.

5.2.4.1 CFG_COMMAND

The CFG_COMMAND statement gives CSC the site specific information needed to implement the *CSC CMD keyin. These statements define parameters and response sequences used to establish a terminal session with the NCS server.

CFG_COMMAND option

This form of the CFG_`COMMAND` parameter allows a site to set terminal session options and response parameters. It has the following format:

`CFG_COMMAND` *`option option-value`*

The following table describes the options.

Option	**Option-value**	**Description / Default**
ACCOUNT	Terminal session account	Account number used when a state based CFG_COMMAND statement response is %ACCOUNT. Default: None
COMPRFX	HSC / Library Station command prefix	If specified, this character precedes all commands submitted by CSC in the CONSOLE state. This limits *CSC CMD input to HCS and Library station

Option	Option-value	Description / Default
		commands. This should match the COMPRFX configuration value given to HSC. Default: None
PASSWORD	Terminal session password	Password when a state based CFG_COMMAND statement response is %PASSWORD. Default: Password from SIGNON configuration parameter
USERID	Terminal session userid	Userid used when a state based CFG_COMMAND statement response is %USERID. Default: Userid from SIGNON configuration parameter

CFG_COMMAND state

This form of the CFG_COMMAND parameter defines actions taken by CSC when specific inputs are received. The actions include sending a response to the server, sending a message to the operator console, and advancing to the next state of session establishment. Following are the statement formats.

```
CFG_COMMAND state prompt response
CFG_COMMAND state,NEXT prompt [response]
CFG_COMMAND state,FAIL prompt console-message
```

In all forms of `CFG_COMMAND`, *`state`* is the active state of session establishment to which this statement applies. Allowable state values are shown in a table that follows.

`Prompt` is the text that CSC must match to perform the indicated action. The prompt text is enclosed in single (‘) or double (“) quotation marks. The prompt may be preceded by * or $ to affect the portion of the received text that can be matched. When the prompt is preceded by an asterisk (*), the entire input line from the server must exactly match the prompt string. When the prompt is preceded by a dollar sign ($), the prompt must match the beginning of the input received from the server. If neither of these is present, the prompt string may occur anywhere within the received text.

In the first two forms, *`response`* is the text that is sent to the server if the indicated prompt is received in the indicated state. If *`state`*`,NEXT` is present, CSC advances the session establishment to the next internal state. If `NEXT` is not present, CSC does not change its internal session establishment state.

The last form is used to inform CSC and the console operator that a condition exists that prevents the establishment of a terminal session. The *`console-`*

`message` is displayed on the console. CSC waits two minutes and restarts the session establishment process.

The following table shows the state specifications.

State Specification	Description
LOGIN SUBMIT CONSOLE	In the indicated state, send the response to the server when the prompt is received. CSC remains in the same state.
LOGIN,NEXT	When the prompt is received in the LOGIN state, CSC sends the response to the server and advances to the SUBMIT state.
SUBMIT,NEXT	When the prompt is received in the SUBMIT state, CSC sends the response to the server and advances to the CONSOLE state.
LOGIN,FAIL CONSOLE,FAIL SUBMIT,FAIL	When the prompt is received in the indicated state, CSC displays the console message text, waits 2 minutes, and then goes to the LOGIN state.

Example

```
CFG_COMMAND USERID CCMDUID
CFG_COMMAND PASSWORD CCMDPWD
CFG_COMMAND LOGIN $”IKJ56700A” %USERID
CFG_COMMAND LOGIN $”IKJ56714A” %PASSWORD
CFG_COMMAND LOGIN $”IKJ56496I” “”
CFG_COMMAND LOGIN,NEXT *'READY' “CONSOLE”
CFG_COMMAND SUBMIT,NEXT “CONSOLE”
CFG_COMMAND LOGIN,FAIL "IKJ56425I" "CMD userid is in use"
CFG_COMMAND LOGIN,FAIL "IKJ56415I" "CMD password has expired"
```

5.2.4.2 CFG_VOLRPT

The CFG_VOLRPT statement gives CSC the site specific information needed to process DO-VOLRPT requests from CSCUI. These statements define parameters and response sequences used to establish a terminal session with the NCS server, execute the volume report utility, and retrieve its output.

CFG_VOLRPT option

This form of the CFG_VOLRPT parameter allows a site to set terminal session options and response parameters. It has the following format:

`CFG_COMMAND` *`option option-value`*

The following table describes the options.

Option	Option-value	Description / Default
ACCOUNT	Terminal session account	Account number used when a state based CFG_VOLRPT statement response is %ACCOUNT. Default: None
COPYPROC	Procedure name	This is the procedure on the NCS server that lists the output from the volume report utility so that CSC can copy it. Default: "rptcopy"
PASSWORD	Terminal session password	Password when a state based CFG_VOLRPT statement response is %PASSWORD. Default: Password from SIGNON configuration parameter
USERID	Terminal session userid	Userid used when a state based CFG_VOLRPT statement response is %USERID. Default: Userid from SIGNON configuration parameter
VOLPROC VOLPROC,HDG	Procedure name	These are the procedures on the NCS server that execute the volume report utility. The first form produces a volume report without headings. The second form produces one with headings. Defaults: "volbat" and "volbath"

CFG_VOLRPT state

This form of the CFG_VOLRPT parameter defines actions taken by CSC when specific inputs are received. The actions include sending a response to the server, sending a message to the operator console, and advancing to the next state of session establishment. Following are the statement formats.

```
CFG_COMMAND state prompt response
CFG_COMMAND state,NEXT prompt [response]
CFG_COMMAND state,FAIL prompt console-message
```

In all forms of `CFG_COMMAND`, *`state`* is the active state of session establishment to which this statement applies. Allowable state values are shown in a table that follows.

`Prompt` is the text that CSC must match to perform the indicated action. The prompt text is enclosed in single (‘) or double (“) quotation marks. The prompt may be preceded by * or $ to affect the portion of the received text that can be matched. When the prompt is preceded by an asterisk (*), the entire input line from the server must exactly match the prompt string. When the prompt is

preceded by a dollar sign ($), the prompt must match the beginning of the input received from the server. If neither of these is present, the prompt string may occur anywhere within the received text.

In the first two forms, *`response`* is the text that is sent to the server if the indicated prompt is received in the indicated state. If *`state`*`,NEXT` is present, CSC advances the session establishment to the next internal state. If `NEXT` is not present, CSC does not change its internal session establishment state.

The last form is used to display a message on the operator console. It allows operator notification of conditions that prevents the establishment of a volume report terminal session. The *`console-message`* is displayed on the console. The volume report request fails with status 19.

The following table shows the state specifications.

State Specification	Description
LOGIN RETRIEVE TRANSFER	In the indicated state, send the response to the server when the prompt is received. CSC remains in the same state.
LOGIN,NEXT	When the prompt is received in the LOGIN state, CSC sends the response to the server. It then sends the volume report utility request defined by the VOLPROC option. Finally, it advances to the RETRIEVE state.
RETRIEVE,NEXT	The volume report utility has finished when the prompt is received. CSC sends the response to the server. It then sends the volume report copy request defined by the COPYPROC option. Finally it advances to the TRANSFER state.
TRANSFER,NEXT	When the prompt is received in the TRANSFER state, CSC sends the response to the server and advances to an internal state that CSC uses to close the terminal session.
LOGIN,FAIL, RETRIEVE,FAIL, TRANSFER,FAIL	When the prompt is received in the indicated state, CSC displays the console message text and terminates the user’s DO-VOLRPT request with a status of 19.

Example

```
CFG_VOLRPT USERID VRPTUID
CFG_VOLRPT PASSWORD VRPTPWD
CFG_VOLRPT LOGIN $”IKJ56700A” %USERID
CFG_VOLRPT LOGIN,FAIL  "IKJ56425I" "Volrpt userid is in use"
CFG_VOLRPT LOGIN $”IKJ56714A” %PASSWORD
CFG_VOLRPT LOGIN,FAIL  "IKJ56415I" "Volrpt password has expired"
CFG_VOLRPT LOGIN $”IKJ56496I” “”
CFG_VOLRPT LOGIN,NEXT *'READY'
CFG_VOLRPT RETRIEVE,NEXT “MAXCC=”
CFG_VOLRPT RETRIEVE,FAIL "IKJ56500I" "Volrpt proc does not exist"
CFG_VOLRPT TRANSFER,NEXT “READY” "logoff"
CFG_VOLRPT TRANSFER,FAIL "IKJ56500I" "VolCopy proc does not exist"
```

5.3 CONFIGURE DEMAND TERMINAL SECURITY

Demand terminal security is a feature that can be controlled by your security officer. The execution of CSC and CDI commands from a demand terminal in console mode (@@CONS), will be regulated by global security privileges. These privileges are established during the product installation process, and can be configured any time after the installation. Authorized security levels include:

1 = Basic
2 = Limited
3 = Full
4 = Display
5 = Response
6 = System Console

Table 5-1 indicates the default values established for each CSC and CDI command during the product installation process.

Table 5-1. Demand Terminal Security Default Values

If your security level is...	**You can execute the following commands for...**	
	CSC	**CDI**
1 or higher	HELP, LEVEL, MEMORY, QUEUES, STATUS	BUFFER, FLAGS, FS, HELP, ID, STATUS
3 or higher	*N/A*	PING
5 or higher	ABORT, ACTIVATE, BRKPT, CLEAR, CYCLE, DEBUG, DISMOUNT, DOWN, EABT, EJECT, ENTER, MOUNT, QUERY, TERM, UP, VENTER	BRKPT, CLR, DN, DUMP, SET, TCP, TERM, UP

5.3.1 Changing Security Levels

Demand terminal security levels can be overridden by editing the CSC$PARAM and CDI$PARAM elements found in the CSC product file. This file is specified in the CSCPARM parameter during the product build process. The default filename is SYS$LIB$*CSC-PARM. To configure these elements, you can use the @ED processor or any other editor that creates ASCII elements. For example, to change the security level for the *CSC QUEUES and *CSC STATUS commands from level 1 to level 2, you must first obtain read and write access to the ACS product file.

The next step is to call the editor and update the CSC$PARAM element. For instance, you might enter the following statement:

```
@ED,U  SYS$LIB$*CSC-PARM.CSC$PARAM
```

Display the entire element, and insert the following parameter at any line within the element:

```
I SEC_LEVEL_2=QUEUES,STATUS
```

This new parameter will override the previously established default security levels for *CSC QUEUES and *CSC STATUS. You can insert as many parameter lines as you want. When you're finished, exit the editor and save your updates. If CSC was active when the changes were made, you will have to terminate and restart CSC before the new security levels take effect.

To change the security level for a *CDI command, you would repeat the above process, but substitute CDI$PARAM for CSC$PARAM. For example, to change the security level for the *CDI HELP and *CDI STATUS commands from level 1 to level 2, call the editor and update the CDI$PARM element. For instance, you might enter the following statement:

```
@ED,U  SYS$LIB$*CSC-PARM.CDI$PARAM
```

Display the entire element, and insert the following parameter at any line within the element:

```
I SEC_LEVEL_2 HELP,STATUS
```

This new parameter will override the previously established default security levels for *CDI HELP and *CDI STATUS. You can insert as many parameter lines as you want. When you're finished, exit the editor and save your updates. If CDI was active when the changes were made, you will have to terminate and restart CDI before the new security levels take effect.

5.3.2 Rules to Consider When Changing Security Levels

When editing the CSC$PARAM and CDI$PARAM elements to override default security levels, consider the following rules:

- When a default security level is overridden, the change is applied globally to all demand terminal users. There is no option to change a security level for an individual user.
- When editing CDI$PARAM, use a space to separate the parameter from the commands. When editing CSC$PARAM, use an equal sign ("=") to separate the parameter from the commands.
- Spaces are not allowed before or after commas.
- If CSC and CDI are active at the time you edit the CSC$PARAM and/or CDI$PARAM elements, it will be necessary to terminate and restart CSC and CDI before the changes take effect.
- If a CSC or CDI command entered on a demand terminal is misspelled or the syntax is incorrect, you will not receive an error or warning.

5.4 CONFIGURE CSCUI

During the COMUS BUILD of the CSC product group, you have the option of installing or suppressing CSCUI. With CSCUI installed, EXCTL, a standalone program, allows you to change CSCUI configurations after the initial software BUILD, including:

- Enabling or disabling a CSCUI Notification Exit or User Request
- Changing the number of users who can register to a particular User Request
- Changing the number of entries that can be queued to, or active on, a particular Notification Exit or User Request
- Changing OS 2200 logging characteristics of CSCUI functions
- Changing the configuration of messages from the CSCUI common bank

In the CSC runstream, the program EXCTL (for "Exit Control") accepts CSCUI-CONFIG as the input parameter element for CSCUI. First, you edit the CSCUI-CONFIG element to customize your environment. This element can be found in the CSCPARM file (the default value for this element is SYS$LIB$*CSC-PARM). These changes take effect the next time you start CSC. Although you can use EXCTL while CSC is active, it is not recommended, because dynamically-configured values are not saved when CSC needs to be restarted.

5.4.1 EXCTL Defaults

Table 5-2 lists the EXCTL defaults. Message options (not listed in the table) are *NONE*.

Table 5-2. EXCTL Defaults

Notification Exit or User Request	Exit #	Max Users	Max Queues	Log options
BEFORE-MOUNT	0	1	50	sends/receives/deletes
MOUNT	1	20	20	sends/receives/deletes
AFTER-EJECT	2	1	50	sends/receives/deletes
AFTER-ENTER	3	1	50	sends/receives/deletes
AFTER-MOUNT	4	1	50	sends/receives/deletes
AFTER-DISMOUNT	5	1	50	sends/receives/deletes
DO-VOLRPT	6	5	5	sends/receives/deletes
DO-ENTER	7	5	5	sends/receives/deletes
INITIATE-EJECT	8	5	5	sends/receives/deletes
SCRATCH/UNSCRATCH	9	5	5	sends/receives/deletes
VOLUME-INFORMATION	10	5	5	sends/receives/deletes
EJECT-COMPLETION	11	1	50	sends/receives/deletes

5.4.2 Using the Exit Control Program (EXCTL)

The EXCTL program reads CSCUI parameters, validates input, updates the CSCUI common bank, and internally reloads the bank. Using EXCTL, you can configure exits and messages.

EXCTL is in the CSC product file, SYS$LIB$*CSC. To execute EXCTL, enter:

```
@SYS$LIB$*CSC.EXCTL[,L|C] bdi
```

Table 5-3 explains the parameters:

Table 5-3. EXCTL Command Parameters

Parameter	Description
L	List input images as they are read in. This option is assumed when the program executes in a batch run.
C	Checkout mode to check the syntax of configuration statements only. This option is always set when the program executes in a demand run.
BDI	Bank Descriptor Index (BDI) value given in the COMUS BUILD of CSC for the common bank CSCUICBA (default 0400302). You *must* specify this value.

You can enter configuration statements manually, or add statements using @ADD. For help on EXCTL statements, enter a "?" at any prompt.

5.4.3 Configuring Notification Exits and User Requests Using EXCTL

The syntax for configuring a Notification Exit or User Request in EXCTL is:

```
CONFIGURE EXIT exit-name WITH MAXIMUM USERS #users ;
   MAXIMUM QUEUES #queues AND LOG log-options
```

Following are descriptions of each of these parameters.

5.4.3.1 *exit-name*

This parameter allows you to specify the exit to configure by *exit-name*. There are two types of exits: *Notification Exits* and *User Requests*.

Notification Exits

Notification Exits include:

- AFTER-EJECT
- AFTER-ENTER
- AFTER-MOUNT
- AFTER-DISMOUNT
- BEFORE-MOUNT
- EJECT-COMPLETION

User Requests

User Requests include:

- DO-ENTER
- DO-VOLRPT
- INITIATE-EJECT
- MOUNT
- SCRATCH/UNSCRATCH
- VOLUME-INFORMATION

5.4.3.2 *#users*

This parameter allows you to specify the number of users that can be registered at one time for a given exit. Table 5-4 shows the maximum number of users for exits.

Table 5-4. Maximum Number of Users for Exits

Exit	Maximum Number of Users
Notification Exits	1
User Requests	5
MOUNT User Request	20

To turn off an exit, set the *#users* to zero (0).

5.4.3.3 *#queues*

This parameter allows you to specify the number of records the common bank will hold for this exit. For notification exits, this is the number of queued entries when no users are available to receive them. The maximum queue size is 50 records. For user request exits, this is the number of requests that can be active at one time. This size cannot be larger than the maximum number of users.

5.4.3.4 *log-options*

The CSCUI common bank can write selected records to the OS 2200 system log file for future retrieval by another run. This parameter allows you to specify the log option for the given exit. Table 5-5 summarizes the log options:

Table 5-5. EXCTL Log Options

Option	Description
ALL	Log records for all Notification Exits and User Requests.
DELETE	Log any notification record discarded because a queue is full.
NONE	No user-selected logs are created.
RECEIVE	Log any CSCUI request that is submitted by CSC to answer a User Request.
SEND	Log any CSCUI request that is submitted with a SEND or SENDW CSCUI request.

If you specify multiple log options, separate each with a space.

Syntax Example

To configure the AFTER-MOUNT Notification Exit with parameters of one user, a queue size of 45 records, and a logging option of send and delete, you would enter:

```
CONFIGURE EXIT AFTER-MOUNT WITH MAXIMUM USERS 1 ;
   MAXIMUM QUEUES 45 AND LOG SEND DELETE
```

or, using the abbreviated form of the syntax:

```
CON EXIT 4 USER 1 QUE 45 LOG SEN DEL
```

5.4.4 Configuring Messages Using EXCTL

EXCTL also allows you to change the configuration of certain messages from the CSCUI common bank, using the syntax:

```
CONFIGURE MESSAGES { ALL | CSC | NONE | RELOAD | USER }
```

Table 5-6 describes the EXCTL message options:

Table 5-6. EXCTL Message Options

Option	Description
ALL	Displays CSC, RELOAD and USER messages as described below.
CSC	Displays a console message when CSC REGISTERs or DEREGISTERs to CSCUI.
NONE	No additional console messages display.
RELOAD	Displays an operator message when a RUN executing the EXCTL program tries to reload the CSCUI common bank. This message asks if the user wants the common bank reloaded by this run, and requires a Y or N response.
USER	Displays a console message when a BATCH or DEMAND program REGISTERs or DEREGISTERs to CSCUI. This message contains the runid and a date/time stamp.

5.4.5 EXCTL Error Messages

If you make a syntax error in EXCTL, you will receive an EXCTL error message. These messages are:

```
* Error invalid command *** Buffer image follows: (display of invalid image) *** End of image ***
```

The first word of the statement is not a command.

```
* Error preceding image ignored
```

This error message indicates a spelling error or misplaced continuation character.

```
* Error invalid exit name specified
```

The *exit-name* specified does not match any of the defined CSCUI exit names.

```
* Error invalid subcommand “xxxxx”
```

A word within the statement does not match any word in the subcommand dictionary. “*xxxxx*” represents the unrecognized word.

```
* Error queue size specified (nn) is illegal max=xx, min=1
```

The specified queue size (nn) is outside the valid range for this notification exit (xx is the maximum queue size allowed).

```
* Error maximum users allowed for a notification exit is nn
* Error maximum users allowed for a user request exit is nn
```

The value for the *#users* parameter is outside the valid range for the type of exit chosen (i.e., Notification Exits > 1; User Requests > 5).

5.5 CONFIGURE THE DYNAMIC TIMER FEATURE

If your WolfCreek LSM configuration requires the dynamic timer feature supported by CSC, add the "W" option to the @XQT statement within the CSC initialization runstream. The configuration requires this CSC feature if the “Exchanges per Hour” option is set to 90.

5.6 COPY UPDATED ELEMENTS TO SYS$LIB$*CSC-PARM

To complete the process, you should copy any updated parameter elements for your configuration to the parameter file SYS$LIB$*CSC-PARM.

5.7 CONFIGURE CARTRIDGE TAPE SUBSYSTEM

NODE statements describe each component in your systems I/O network. The I/O network can be expressed as a hierarchical relationship of interconnected hardware components. Devices, or cartridge tape drives, are at the lowest level, and the central processor, or CPU, is at the highest level. NODE statements are used to completely describe each component in the network and the path or paths to the next component in the hierarchy. Before you start an EXEC generation, you must first write a set of NODE statements to configure the channel, control units, and transport units within your systems I/O network. These statements will then be input to the EXEC generation. You must define the cartridge tape subsystems that will be used in your ACS.

5.8 SERVER CONFIGURATION CONCERNS

5.8.1 TCP/IP Issues

CSC and the server software create TCP connections to send each request and response. When the server attempts to open a connection for a response, it may try for several minutes before giving up. The server software performs connection opens sequentially. If one connection cannot be opened, it will be several minutes before subsequent opens are attempted. A common cause of connection open failures is that the active communication interface program or the system on which CSC is executing is taken down before all responses have been received. If this occurs with multiple responses outstanding then the server could spend a long time trying to establish connections.

On the Solaris-based server the time allowed by the server to complete a connection open can be changed by setting the tcp_ip_abort_cinterval parameter of /dev/tcp. This is described in PLE 16930610.

After each of these connections between CSC and the server is used, it is closed. The close process includes a wait period that is timed to let any remaining communication frames clear the network. The default setting assumes a worldwide network that includes some slow links.

On the Solaris-based server, the default timed wait period is 3 minutes. When many requests and responses are exchanged, the server will have many connections in the timed wait state. This uses server resources and can affect server performance. The timed wait period on the Solaris-based server can be changed by setting the tcp_close_wait_interval parameter of /dev/tcp.

The following procedure sets these parameters to values that have been tested under heavy request traffic and adverse communication conditions.

1. Log on to the Solaris-based server as the super user
2. ndd -set /dev/tcp tcp_ip_abort_cinterval 15000
3. ndd -set /dev/tcp tcp_close_wait_interval 10000
4. Log off of the Solaris-based server

5.8.2 Additional Solaris-Based Server Concerns

It is recommended that the CSI tuning variable for number of CSI retries be set to 2 in the ACSLS configuration.

If your site uses a dual-LAN configuration, it is recommended that a csc_ip_switch.dat file be created on the server to enable the ACSLS dual path option. Refer to the *ACSLS System Administrator's Guide*, "Managing a Dual-LAN Client Configuration" section for more information on the dual path option and on creating the csc_ip_switch.dat file.

5.8.3 Additional NCS Server Concerns

The CSCUI volume report and the console command feature both require a terminal session between CSC and the NCS server. The CSC configuration parameters described in the NCSRPARAM section (reference to NCS$PARAM section) provide signon information for these sessions. On the NCS server, these userids must be established with the appropriate privileges.

The volume report uses three JCL procedures on the NCS server; two for executing the volume report utility and one to transfer the output to CSC. These procs must be created and accessible to the userid that will perform the volume report. Samples of these procs are contained in the NCS-PROCS element in the CSC installation file.

6. STARTING CSC & OPTIONAL VERIFICATIONS

This chapter describes how to start CSC. Make certain that you've completed the installation and configuration procedures described in previous chapters *before* you attempt to start CSC.

In addition, this chapter describes optional procedures you can use to verify that CSC and CDI are correctly configured and operational.

6.1 STARTING CSC

Once you've installed and configured CSC as previously described, you're ready to start CSC.

Use these procedures to start CSC:

1. TERMINATE the CDI run (if already running).
2. TERMINATE the CSC run (if already running).
3. If, during the COMUS BUILD of CSC, you opted to use the CSC User Interface (CSCUI), you must also reload the following common banks:
   - CSCUICBA
   - CSCUICBB
4. START the CDI run, if using the CDI interface.
5. START CMS, if using the CMS interface and CMS is not already running.
6. START CPCOMM, if using the CPCOMM interface and CPCOMM is not already running.
7. START the CSC run.

The FAST-STARTUP feature is enabled by default. To disable it, START the CSC run with a SET value of 0100 (`ST CSC,0100`).

6.2 OPTIONAL VERIFICATIONS

6.2.1 CSC$PARAM Configuration Verification

After the CSC$PARAM configuration element has been updated, a utility can be run to verify the new PATH and INTERFACE configuration statements. This utility is CSCPARM/VERIFY and is located in the CSC product file after CSC has been installed. CSCPARM/VERIFY will verify that the CSC$PARAM element has at least one INTERFACE statement defined, and that each PATH statement has a corresponding INTERFACE.

Additionally, if a CDI$PARAM element exists, CSCPARM/VERIFY will verify that the CDI INTERFACE is defined and that each PATH using the CDI interface has a corresponding ROUTE statement in CDI$PARAM. Also, the CPA and server addresses in each PATH using the CDI interface will be verified against the corresponding ROUTE statement in CDI$PARAM.

This utility can be executed from a demand run by entering:

```
@ADD SYS$LIB$*CSC.CSCPARM/VERIFY
```

This utility will add the CSC$PARAM and CDI$PARAM from the CSC Configuration file and display error messages if any errors are encountered. The following error messages describing the error could be displayed:

```
*CONFIG ERROR: NO INTERFACE STATEMENTS
```

```
*CONFIG ERROR: NO PATH STATEMENTS

**CONFIG ERROR: INTF [INTERFACE,I,1,1] HAS NO PATH STATEMENTS

*CONFIG ERROR: NO CDI$PARAM ROUTE FOR PATH pathname

*CONFIG ERROR: INTERFACE STATEMENT FOR CDI DOES NOT EXIST
```

If CSCPARM/VERIFY finds errors in the CSC$PARAM element, the following message is displayed on the terminal and on the system console:

```
*CSC$PARAM CONTAINS ERRORS
```

If no errors are encountered, the following message is displayed:

```
*CSC$PARAM VERIFIED
```

6.2.2 *CDI PING Command

After completing the CSC installation, if CDI is configured, you may want to verify that CDI is functioning properly and that it can communicate with the server. You can do this using the *CDI PING command.

The *CDI PING command lets you probe any server (configured in the CDI$PARAM element) across the LAN to verify that it is reachable. The server simply echoes the request with a timestamp. CDI will calculate the average response for the request.

If PING executes successfully, this verifies that both the hardware and software connectivity to the server are available and functioning.

To “PING” the server, first verify that CDI is running. Next, enter the following command at the system console:

```
*CDI PING IP-address
```

where *IP-address* is the server’s Internet Protocol address.

If PING responds with the message, “PING Timeout,” then the connection to the server is not available. Check the server configuration to verify that the IP addresses are correct.

If PING is successful, it responds with a message similar to:

```
PING RESPONSE TOOK .50 SECS, SEQ#12
PING SUCCESSFUL.
```

The *SEQ#* displayed in the example shows that there have been a total of 12 PING commands issued since the start of CDI.

INDEX

EFFECTIVE PAGES

Manual:	CSC Installation Guide 313471401
Issue Date:	July 2001
Reissue Date:	

This document has 96 total pages, including:

Title			
Copyrights, Disclaimers & Trademarks			
Contents, Tables & Figures			
Preface	xi	thru	xiv
Chapter 1:	1-1	thru	1-4
Chapter 2:	2-1	thru	2-18
Chapter 3:	3-1	thru	3-12
Chapter 4:	4-1	thru	4-10
Chapter 5:	5-1	thru	5-26
Chapter 6:	6-1	thru	6-4
Index	Index-1	thru	Index-6
Effective Pages			
Reader's Comment Form			
Business Reply Mailer			