BASEstar™ Classic DAS for Allen-Bradley INTERCHANGE™ Software

Installation and User's Guide

Order Number: AA-R213B-TE

April 2000

This manual describes how to install and use the DAS for Allen-Bradley INTERCHANGE software for BASEstar Classic on OpenVMS.

Revision/Update Information:	This is a revised document.
Operating System and Version:	OpenVMS/Alpha Version 6.1
Operating System and Version:	OpenVMS/VAX Version 5.5-2
Interface Software and Version:	BASEstar Classic Version 3.4
Allen-Bradley Software:	INTERCHANGE Software 6.0.2
Software Version:	BASEstar Classic DAS for Allen-Bradley INTERCHANGE Software, Version 3.4A

Compaq Computer Corporation
Houston, Texas

April 2000

© 1992 Compaq Computer Corporation.

COMPAQ, the Compaq logo, VMS, and VAX Registered in U.S. Patent and Trademark Office.

BASEstar and OpenVMS are trademarks of Compaq Information Technologies Group, L.P. in the United States and/or other countries.

PLC, PLC-2, PLC-3 and PLC-5 are registered trademarks of Allen-Bradley Company Inc.

INTERCHANGE, PLC-5/250, SLC, SLC 500, SLC 5/01, SLC 5/02, SLC 5/03, SLC 5/04, SLC 5/05, Pyramid Integrator, Data Highway, DH+ and Data Highway Plus are trademarks of Allen-Bradley Company Inc.

All other products mentioned herein may be trademarks or registered trademarks of their respective companies.

Confidential computer software. Valid license from Compaq or authorized sublicensor required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentaion, and Technical data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is subject to change without notice.

This document is available on CD–ROM.

This document was prepared using DECdocument, Version 3.3.

Contents

Preface vii

1 Overview

1.1 Description 1–1
1.2 Device Communications 1–1
1.3 Supported Functions and Devices 1–5
1.4 Errors messages and returned status codes 1–5

2 Installing and Configuring the DAS

2.1 Installation Requirements 2–1
2.1.1 Hardware 2–1
2.1.2 Software 2–2
2.1.3 Additional Software 2–2
2.1.4 Disk Space 2–3
2.1.5 Preparation for installing over old PI or DTL DAS 2–3
2.2 Installation 2–4
2.2.1 Files Created During Installation 2–7
2.2.2 Installation Messages 2–9
2.3 Postinstallation Tasks 2–10
2.3.1 Modify TCP/IP for OpenVMS UCX parameters 2–11
2.3.2 Editing the Configuration File 2–11
2.3.2.1 Editing Type Records 2–11
2.3.2.2 Editing Path Records 2–15
2.3.2.3 Editing Device Records 2–17
2.3.3 Setting the BASEstar Classic Message Port Quota Parameter 2–19
2.3.4 Setting the SPT Block Parameter 2–19
2.3.5 Setting DAS-Specific Parameters 2–20
2.3.5.1 ILAN$DTLSRV_PRIORITY 2–20
2.3.5.2 ILAN$DTL_BACKLOG 2–20
2.3.5.3 ILAN$DTL_CYCLE_TIME 2–21
2.3.5.4 ILAN$DTL_DWNL_FLAGS 2–21
2.3.5.5 ILAN$DTL_MAX_ELEMENTS 2–21
2.3.5.6 ILAN$DTL_MAX_UNSOL_ELEMENTS 2–21
2.3.5.7 ILAN$DTL_SCMP_FLAGS 2–22
2.3.5.8 ILAN$DTL_TIMEOUT 2–22
2.3.5.9 ILAN$DTL_UDC_TIMEOUT 2–22
2.3.5.10 ILAN$DTL_UPL_FLAGS 2–22
2.3.5.11 ILAN$DTL_DUP_EI_ID 2–23
2.3.6 Setting the DTL Server Process Quotas 2–23
2.3.7 Setting Up Plant-Floor Equipment 2–23
2.4 Tracing Device Communications 2–24
2.5 Failures During Product Use 2–24

3 Using the DAS

3.1 Accessing DAS Functions 3–1
3.2 Supported Functions 3–2
3.2.1 Read and Write Data Functions 3–2
3.2.1.1 PLC-5/250 and PLC-5 Addressing 3–2
3.2.1.2 SLC Addressing 3–3
3.2.1.3 PLC-3 Addressing 3–4
3.2.1.4 PLC-2 Addressing 3–5
3.2.1.5 Reading and Writing Data 3–5
3.2.2 Read Status Function 3–6
3.2.3 Upload and Download Functions 3–8
3.2.3.1 DTL_DOWNLOAD 3–9
3.2.3.2 DTL_UPLOAD 3–10
3.2.4 Loop Diagnostic Function 3–11
3.2.5 Simple Compare Function 3–12
3.3 Supported Data Types 3–13
3.3.1 Structured Data 3–13
3.3.2 Restrictions to Data Formats 3–14
3.4 Automatic Data Collection 3–15
3.4.1 Unsolicited Data Collection 3–16
3.4.2 Pollsets 3–19
3.5 Supporting Utilities 3–20
3.5.1 Stop Server Utility 3–20
3.5.2 Supporting Utilities for Allen-Bradley 6200 Series Software 3–21
3.5.2.1 Import 6200 File Utility 3–21
3.5.2.2 Export 6200 File Utility 3–21

A Allen-Bradley PLC-5/250 Addressing

B Allen-Bradley PLC-5 Family Addressing

C Allen-Bradley SLC Family Addressing

D Logged Messages

D.1 NI Logged Messages D–1
D.2 PE Logged Messages D–2
D.3 DTL Server Logged Messages D–3

Index

Examples

2–1 Type Configuration 2–11
2–2 Path Configuration for EI Network Interface 2–16
2–3 Path Configuration for PI MicroVAX 2–16
2–4 Device Configuration 2–17
3–1 READ DATA Output 3–6
3–2 WRITE DATA Output 3–6
3–3 READ STATUS Output for PLC-5/250 3–7

3–4 READ STATUS Output for PLC-5 3–7
3–5 READ STATUS Output for SLC 3–8
3–6 READ STATUS Output for PLC-3 3–8
3–7 READ STATUS Output for PLC-2 3–8
3–8 UPLOAD and DOWNLOAD Input 3–9
3–9 Loop Output 3–12
3–10 Stop Server Utility 3–20
3–11 Import 6200 File Utility 3–21
3–12 Export 6200 File Utility 3–21

Figures

1–1 DAS Communications 1–2
1–2 PI MicroVAX Configuration 1–3
1–3 EI Network Interface Configuration 1–4
3–1 Valid Unsolicited ID Configuration 3–18
3–2 Invalid Unsolicited ID Configuration 3–19

Tables

2–1 Disk Space Requirements 2–3
2–2 DAS Files Created (EI Network Interface Option) 2–8
2–3 DAS Files Created (PI MicroVAX Option) 2–8
2–4 Files Modified 2–9
2–5 Path Parameters 2–17
2–6 SPT Static Block Sizes 2–19
2–7 SPT Dynamic Block Sizes 2–19
3–1 Supported Devices and Functions 3–2
3–2 PLC-5 Supported Section Files 3–3
3–3 SLC Supported Section Files 3–4
3–4 PLC-3 Supported Section Files 3–5
3–5 Read and Write Data Interface Access 3–5
3–6 Read Status Interface Access 3–7
3–7 Upload and Download Interface Access 3–9
3–8 Download Qualifiers 3–10
3–9 Upload Qualifiers 3–11
3–10 Loop Interface Access 3–11
3–11 Simple Compare Interface Access 3–12
3–12 Simple Compare Qualifiers 3–13
3–13 Data Formats Supported for Structured Data 3–14
3–14 PLC-5 MSG Instruction for Unsolicited Data 3–16
3–15 PLC-5/250 MSG Instruction for Unsolicited Data 3–17
A–1 Supported PLC-5/250 Addresses and Data Types A–1
B–1 Supported PLC-5 Family Addresses and Data Types B–1
C–1 Supported SLC Family Addresses and Data Types C–1

Preface

This document describes how to install and use the BASEstar Classic DAS for Allen-Bradley INTERCHANGE software.

Intended Audience

This document is intended for system managers who must set up and maintain the following:

- BASEstar Classic software
- BASEstar Classic DAS for Allen-Bradley INTERCHANGE software

This document is also intended for application programmers who develop plant-floor management software layered on the BASEstar Classic software.

Readers of this document should have knowledge of:

- OpenVMS operations and administration
- OpenVMS application software
- BASEstar Classic software
- Site-specific installation requirements

In addition, knowledge of the Allen-Bradley devices is required.

Document Structure

This document is organized as follows:

- Chapter 1, Overview, is an overview of the DAS for AB INTERCHANGE software.
- Chapter 2, Installing and Configuring the DAS, provides the information needed to install and configure the DAS for AB INTERCHANGE software.
- Chapter 3, Using the DAS, describes the supported functions for Allen-Bradley devices and how to access those functions.
- Appendix D, Logged Messages, describes messages logged to the BASEstar Classic history file for the DAS for AB INTERCHANGE software.

Associated Documents

For information on installing and configuring a BASEstar Classic system, consult the following documents:

- *BASEstar Classic Release Notes*
- *BASEstar Classic Installation Guide*
- *BASEstar Classic Configuration and Tuning Guide*

For instructions on how to use BASEstar Classic software functions, consult the following documents:

- *BASEstar Classic Introduction*
- *BASEstar Classic Command Line Interface User's Guide*
- *BASEstar Classic Menu Interface User's Guide*

For more information on developing software to integrate manufacturing applications and equipment, consult the following documents.

- *BASEstar Classic Introduction to Callable Services*
- *BASEstar Classic Application Programming Interface Reference Guide*
- *BASEstar Classic Guide to Writing Device Access Software*
- *BASEstar Classic Guide to Writing a Database Server*
- *BASEstar Classic DEC Rdb Database Server Guide*

For information on Allen-Bradley INTERCHANGE software and devices, consult the following documents:

- *Pyramid Integrator Design Manual*
- *Pyramid Integrator Start-up and Integration Manual*
- *INTERCHANGE Software User Manual*
- *INTERCHANGE Software Concepts Manual*
- *INTERCHANGE Software Reference Manual*
- *Pyramid Integrator Addressing Reference*
- 6200 *PLC-5/250 Programming Software* Documentation Set.
- *PLC-5 Family Addressing Reference*
- 6200 *PLC-5 Programming Software* Documentation Set.

Note

Consult with your Allen-Bradley Sales representative for the most up to date document related to the topic.

Conventions

This document uses the following conventions:

Boldface	Highlights user input within textual descriptions.
Return	Press the key labeled Return. Unless otherwise specified, press Return after entering a command or responding to a prompt.
Enter	Type the words or symbols described and press Return.

1
Overview

This chapter provides an overview of the BASEstar Classic DAS for Allen-Bradley INTERCHANGE software. It also briefly describes Allen-Bradley device communications, and the supported functions for the DAS for AB INTERCHANGE software.

1.1 Description

The DAS for AB INTERCHANGE software allows you to access Allen-Bradley devices using BASEstar Classic device connection management capabilities. Device connection management is the device connection and control component of BASEstar Classic for OpenVMS that provides generic communication with plant equipment, allowing the user to:

- Configure device definitions
- Collect data from plant equipment
- Control device operation

Using the DAS for AB INTERCHANGE software, users or applications can perform a variety of device access functions for the Allen-Bradley devices.

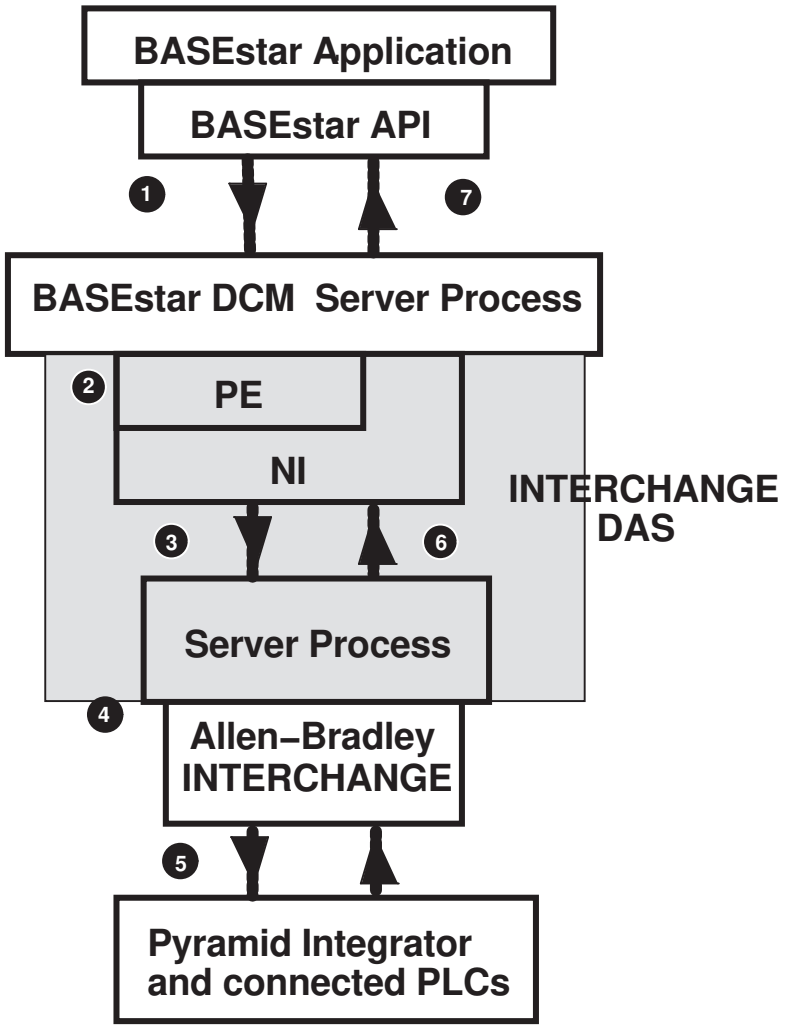
1.2 Device Communications

The DAS for AB INTERCHANGE software consists of a protocol emulator (PE), a network interface (NI) and an INTERCHANGE server process. The PE and NI work together to provide device-specific communications for BASEstar Classic device connection management functions. The INTERCHANGE server process (DCM_DTL$SRV.EXE) implements these BASEstar Classic device connection management functions through Allen-Bradley INTERCHANGE software function calls.

The PE translates BASEstar Classic device connection management generic services into a device-understandable format. The PE also converts device-specific protocol into a BASEstar Classic device connection management format.

The NI communicates the requests and data translated by the PE. The NI works directly with the INTERCHANGE server process to send and receive commands which are translated into INTERCHANGE DTL library calls. These INTERCHANGE DTL library calls send data to and receive data from plant-floor devices. Figure 1–1 shows how the DAS for AB INTERCHANGE software facilitates communications between BASEstar Classic device connection management software and INTERCHANGE software and the device.

Figure 1–1 DAS Communications



❶ Requests typically are initiated by a BASEstar Classic application making an access call through the BASEstar Classic API. This call results to a service request being sent to the BASEstar Classic DCM server process, ILAN$DEVSRV.

❷ ILAN$DEVSRV responds to the request by calling the appropriate PE routine. The PE builds a INTERCHANGE server request packet and calls the NI.

❸ The NI sends the request packet in a BASEstar Classic message to the INTERCHANGE server process, DCM_DTL$SRV.

❹ The server process makes the appropriate DTL API call based on the information in the request message packet.

❺ The desired DTL I/O is performed and status is returned to the server process through INTERCHANGE.

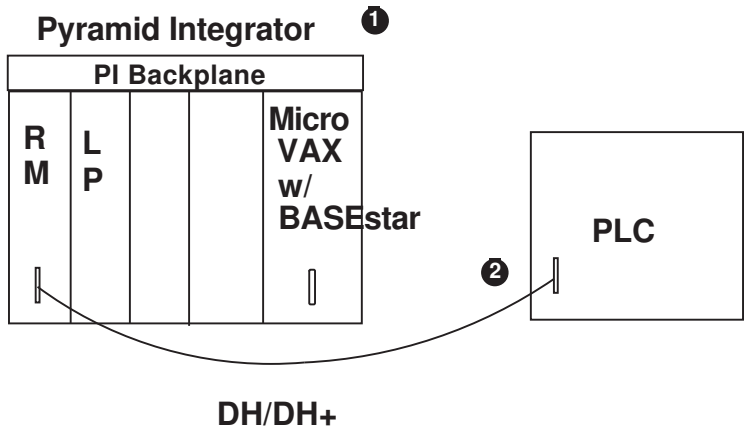
❻ The server process creates a response message and sends it back to ILAN$DEVSRV where it is received by the NI and processed by the PE.

❼ ILAN$DEVSRV passes the response back to the process which initiated the request

The DAS for AB INTERCHANGE software uses the Allen-Bradley INTERCHANGE software products as the means to access Allen-Bradley PLC-5/250 on a Pyramid Integrator chassis, Allen-Bradley PLC processors on a Data Highway/Data Highway Plus (DH/DH+) link off of a Pyramid Integrator chassis or directly connected Allen-Bradley PLC-5/XXE processors. Two versions of INTERCHANGE software are supported by the DAS for AB INTERCHANGE software. They are INTERCHANGE Software for the PI MicroVAX and INTERCHANGE Software for OpenVMS.

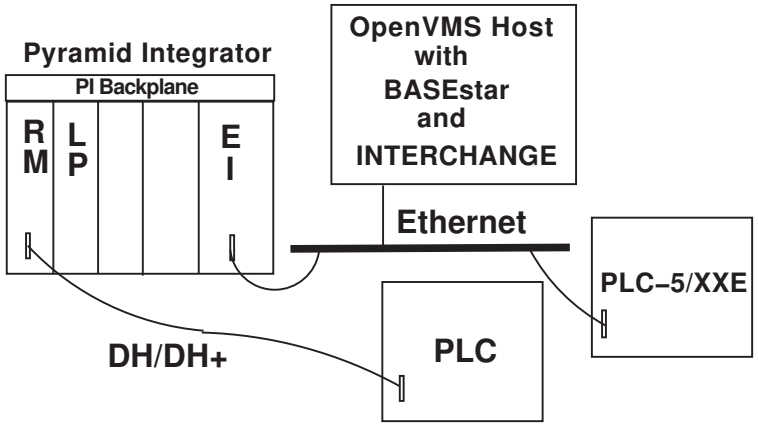
The INTERCHANGE Software for the PI MicroVAX product runs only on an Allen-Bradley VAX Integration Processor. This is a special OpenVMS/VAX system which fits into the back-plane of a Pyramid Integrator Chassis. INTERCHANGE Software for the PI MicroVAX communicates through the Pyramid Integrator Back Plane Interface (BPI) to the Resource Manager (RM) module. Access to remote PLC's is through Data Highway/Data Highway Plus ports on either the Resource Manager or a KA communications adapter in the Pyramid Integrator. Figure 1–2 shows a typical INTERCHANGE Software for the PI MicroVAX configuration.

Figure 1–2 PI MicroVAX Configuration



The INTERCHANGE Software for OpenVMS product runs on any standard OpenVMS system. INTERCHANGE Software for OpenVMS communicates using TCP/IP protocol, through an Ethernet Interface Module, to a Pyramid Integrator chassis or PLC-5/XXE processor. When connecting to a Pyramid Integrator chassis, remote PLC's are linked through the Resource Manager or KA communications adapter modules in the Pyramid Integrator chassis. Communication to these remote PLC processors is through Allen-Bradley's Data Highway/Data Highway Plus network. PLC-5 models with an "E" extension have an Ethernet Interface Module that allows direct connection to the ethernet. Figure 1–3 shows a typical INTERCHANGE Software for OpenVMS configuration.

Figure 1–3 EI Network Interface Configuration



INTERCHANGE Software for OpenVMS can support applications with up to 40 connections to Ethernet Interface Modules. Ethernet Interface Modules can support up to 64 connections. Because the DAS for AB INTERCHANGE software is considered an INTERCHANGE application, these restrictions effect how the DAS coexists with other INTERCHANGE applications within a INTERCHANGE /BASEstar Classic network.

Because the two versions of INTERCHANGE software are mutually exclusive on an OpenVMS/VAX system, the DAS for AB INTERCHANGE software asks the user at installation time to select which option to install.

1.3 Supported Functions and Devices

BASEstar Classic device connection management offers a number of device access functions. To perform a particular operation, the DAS PE must support that function. Device connection management functions are accessed through the following BASEstar Classic interfaces:

- Command line interface (CLI)
- Menu interface
- Application programming interface (API)

The DAS for AB INTERCHANGE software supports Allen-Bradley PLC-5, PLC-3, PLC-2 and SLC family devices and Allen-Bradley model PLC-5/250

Not all functions are supported for the model PLC-5/250 or PLC-3, PLC-2 or SLC family devices or the INTERCHANGE Software for the PI MicroVAX DAS option.

Both options of the DAS for AB INTERCHANGE software support the following device connection management functions:

- Read data from a specific address in a device's memory
- Write data to a specific address in a device's memory
- Receive unsolicited data from a device
- Read the status for a device
- Loop PE
- Loop NI

In addition to the above, the EI Network Interface option supports the following functions for PLC-5 family (excluding the 5/250) devices:

- Upload the contents of a device's memory to an OpenVMS file
- Download an OpenVMS file to a device's memory
- Perform a simple compare of device files

NOTE: Simple compare is not provided on OpenVMS/Alpha systems.

The EI Network Interface option also provides utilities which import and export OpenVMS device files from and to Allen-Bradley's 6200 Programming Software environment.

For more information about supported functions and devices, refer to Chapter 3, Using the DAS.

1.4 Errors messages and returned status codes

The return values generated by DAS for AB INTERCHANGE software are either standard ILAN$_ values or are from Allen-Bradley INTERCHANGE software. The return values generated by DTL have the facility code of DTL_. For more information about these messages, refer to Allen-Bradley documentation or the file SYS$LIBRARY:DTL.H.

To enable message translation of these DTL_ messages for each process, enter the following command:

```
$ SET MESSAGE SYS$MESSAGE:DTLMSG.EXE
```

2
Installing and Configuring the DAS

This chapter provides the information you need to install the DAS for AB INTERCHANGE software and configure your system.

2.1 Installation Requirements

Review the following hardware and software requirements to ensure that your system is prepared for the BASEstar Classic DAS installation.

Note

Back up the disks on your system before installing this software. This will provide a method to restore your system in the event of an installation problem. The procedure for backing up disks is described in the *OpenVMS System Management Utilities Reference Manual*.

2.1.1 Hardware

The hardware requirements for the DAS for AB INTERCHANGE software are the same as those for the BASEstar Classic device connection management services. For specific hardware requirements, see the *BASEstar Classic Installation Guide*.

Additional hardware required for configuring the BASEstar Classic DAS for Allen-Bradley INTERCHANGE software includes one of the following:

- Allen-Bradley Pyramid Integrator Chassis with:
  - Ethernet Interface Module
  - Resource Manager Module
  - Optional KA communications adapter modules
  - Optional LP processor module(s)

  or:
  - Allen-Bradley MicroVAX Information Processor
  - Resource Manager Module
  - Optional KA communications adapter module(s)
  - Optional LP processor module(s)
- Allen-Bradley PLC-5/XXE processor
- Allen-Bradley SLC-5/05 processor
- Optional Allen-Bradley PLC-5 system(s)
- Optional Allen-Bradley PLC-3 system(s)
- Optional Allen-Bradley PLC-2 system(s)

- Optional Allen-Bradley SLC system(s)

Refer to related Allen-Bradley documentation or your Allen-Bradley representative for more complete and current hardware configurations and revision levels.

2.1.2 Software

Before installing the DAS for AB INTERCHANGE software, the following software must already be installed:

- OpenVMS Version 5.5-2 or higher (VAX)
- OpenVMS Version 6.1 or higher (Alpha)
- VAX/FMS Version 2.3 or Version 2.4 (required only when the menu system will be used) (The Menu System is available only on OpenVMS/VAX systems.)
- BASEstar Classic for OpenVMS, Version 3.4 (VAX)

  BASEstar Classic Library System is required only if simple file compare will be used. (BASEstar Classic Library System is available only on OpenVMS /VAX systems.)

If the BASEstar Classic software is not already installed on your system, refer to the *BASEstar Classic Installation Guide*.

Note

Before using this product on a system, you must first register a License Product Authorization Key (License PAK) using the License Management Facility (LMF). For more information about the License Management Utility, refer to the *License Management Utility Manual* for OpenVMS.

2.1.3 Additional Software

For the DAS for AB INTERCHANGE software to function correctly, the following software must also be installed for each installation option.

For device access over Ethernet:

- Allen-Bradley INTERCHANGE Software for OpenVMS 6.0.2
- TCP/IP Services for OpenVMS (UCX) 2.0E

Note

Functionality is limited for SLC processors when using older versions of INTERCHANGE Software for OpenVMS. Consult with your Allen-Bradley sales representative for the most up to date version of INTERCHANGE Software for OpenVMS and the functionality supported.

For device access from an Allen-Bradley Pyramid Integrator MicroVAX information processor :

- Allen-Bradley INTERCHANGE Software for the PI MicroVAX 3.1.1

2.1.4 Disk Space

Table 2–1 lists the disk space required to install the DAS for AB INTERCHANGE software. The space requirements are approximations; actual sizes may vary depending on your system environment, configuration, and software options installed.

Table 2–1 Disk Space Requirements

	Approximate Space Requirements (Blocks)	
Usage	**Ethernet option**	**PI MicroVAX option**
Peak (during installation)	5600 (VAX)	5600
	5000 (Alpha)	
Net (after installation)	500 (VAX)	550
	1000 (Alpha)	

2.1.5 Preparation for installing over old PI or DTL DAS

If you are installing the BASEstar Classic DAS for Allen-Bradley INTERCHANGE software on a system where either the BASEstar Classic device access software for Allen-Bradley Pyramid Integrator (PI DAS) or BASEstar Classic device access for Allen-Bradley Data Table Library Software (DTL DAS), then steps must be taken to deactivate these products before continuing the installation.

To deactivate the PI DAS do the following:

1. Backup BASEstar Classic definitions to a command file (refer to BASEstar Classic documentation on methods)
2. Disable all devices of type/path PI
3. Disconnect any points, delete any phypoints and pollsets associated with these PI devices
4. Delete the PI devices.
5. Stop the BASEstar Classic system
6. Install the DAS for AB INTERCHANGE software using the PI MicroVAX option.
7. Perform post-installation tasks described later in this document
8. Define the old PI devices as new dtlbpi-path devices.
9. Modify the definition command file to reflect device definition changes and re-establish the BASEstar Classic definitions.

To deactivate the Network DTL DAS do the following:

1. Backup BASEstar Classic definitions to a command file (refer to BASEstar Classic documentation on methods)
2. Disable all devices of type/path DTL
3. Disconnect any points, delete any phypoints and pollsets associated with these DTL devices
4. Delete the DTL devices.

5. Stop the BASEstar Classic system
6. Install the DAS for AB INTERCHANGE software using the OpenVMS (Ethernet) option.
7. Perform post-installation tasks described later in this document
8. Define the old DTL devices as new DTLNET-path devices.
9. Modify the definition command file to reflect device definition changes and re-establish the BASEstar Classic definitions.

2.2 Installation

When your system meets all hardware and software requirements, you can install the DAS for AB INTERCHANGE software. The installation takes from 5 to 10 minutes, depending on the system load and configuration.

Install the DAS for AB INTERCHANGE software by using the following steps:

1. Log in to a privileged system manager's account.
2. Set the default directory to SYS$UPDATE:

   ```
   $  SET DEFAULT SYS$UPDATE
   ```

3. Invoke VMSINSTAL:

   ```
   $  @SYS$UPDATE:VMSINSTAL DCM_DTLVVA034 ddcu:
   ```

   where:

   - **DCM_DTLVVA034** argument is the kit name
   - **034** portion of the name is the version number
   - **ddcu** argument represents the name of the device on which the installation media is mounted, where:
     - **dd** is the device code
     - **c** is the controller designation
     - **u** is the unit number

VMSINSTAL prompts you for information during the installation. Note that DECnet software does not need to be running to perform the installation procedure. The following is an example of the output from the installation:

```
$ @sys$update:vmsinstal DCM_DTLVVA034 mua0:


        VAX/VMS Software Product Installation Procedure V5.5-2

It is 28-DEC-1999 at 14:15.

Enter a question mark (?) at any time for help.

* Are you satisfied with the backup of your system disk [YES]?

The following products will be processed:

  DCM_DTLVVA V3.4

        Beginning installation of DCM_DTLVVA V3.4 at 14:15

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.
```

```
   Copyright 1992 Compaq Computer Corporation

   Confidential computer software.  Valid license from Compaq or authorized
   sublicensor required for possession, use or copying.  Consistent with FAR
   12.211 and 12.212, Commercial Computer Software, Computer Software
   Documentation, and Technical Data for Commercial Items are licensed to the
   U.S. Government under vendor's standard commercial license.

   BASEstar Classic DAS for Allen-Bradley INTERCHANGE Software
   installation procedure.

   Checking for a valid license...


        Product:      DAS-DTL-CL
        Producer:     DEC
        Version:      3.4
        Release Date: 13-DEC-1996

* Does this product have an authorization key registered and loaded? Y

   Now checking OpenVMS version...

   Now checking that BASEstar Classic is installed...


   The BASEstar Classic DAS for Allen-Bradley INTERCHANGE Software
   installs as two variants:

   1) Support for INTERCHANGE Software for OpenVMS.  This communicates
   to Allen-Bradley devices from an OpenVMS system through an Ethernet
   Interface.

   2) Support for INTERCHANGE Software for PI MicroVAX.  This option
   runs only on a PI MicroVAX embedded in a Pyramid Integrator Chassis.
   Communication to Allen-Bradley devices is over the backplane
   interface (BPI).

   These two options are mutually exclusive and require the
   corresponding type of Allen-Bradley INTERCHANGE Software to be
   installed.

* Do you wish to install the Ethernet version of the DAS? Y
%DCM_DTLVVA-I-DTLNET, Installing Ethernet version of the DAS...

   Now checking INTERCHANGE software version...

   Now checking disk space...

   Now checking BASEstar Classic message port quota...

   The current value for BCC$MAX_PORT_QUOTA of 50000 is too low
   for the BASEstar Classic DAS for Allen-Bradley INTERCHANGE Software.

* Value to set BCC$MAX_PORT_QUOTA to [100000]:

   The new BCC$MAX_PORT_QUOTA will take effect the next time that BASEstar
   Classic is restarted.  Restart BASEstar Classic software before attempting
   to use this DAS.

* Do you want to purge files replaced by this installation [YES]?
* Do you want to run the IVP after the installation [YES]?

   The installation procedure has no further questions to ask and will
   complete in 5 to 10 minutes depending on the system and system load.

%VMSINSTAL-I-RESTORE, Restoring product save set B ...

   The configuration template file for DTL support, DCM_DTL_CONFIG.TEMPLATE,
   is used to define the Allen-Bradley paths, types, and devices.  Edit this
   file, as necessary, to reflect your specific site configuration.
```

```
   During installation it will be placed in the directory BCC$SYSDATA.

   Now linking executable images with INTERCHANGE software...

   The BASEstar Classic parameter ILAN$DTL_TIMEOUT has just been created.

Parameter ILAN$DTL_TIMEOUT, type Value.
    Current value is 1000,  Default value is 1000.
    Minimum value is 100,  Maximum value is 600000.

   The BASEstar Classic parameter ILAN$DTL_MAX_ELEMENTS has just been created.

Parameter ILAN$DTL_MAX_ELEMENTS, type Value.
    Current value is 500,  Default value is 500.
    Minimum value is 50,  Maximum value is 100000.

   The BASEstar Classic parameter ILAN$DTL_BACKLOG has just been created.

Parameter ILAN$DTL_BACKLOG, type Value.
    Current value is 5,  Default value is 5.
    Minimum value is 1,  Maximum value is 40.

   The BASEstar Classic parameter ILAN$DTL_CYCLE_TIME has just been created.

Parameter ILAN$DTL_CYCLE_TIME, type Value.
    Current value is 70,  Default value is 70.
    Minimum value is 5,  Maximum value is 1000.

   The BASEstar Classic parameter ILAN$DTL_MAX_UNSOL_ELEMENTS has just
   been created.

Parameter ILAN$DTL_MAX_UNSOL_ELEMENTS, type Value.
    Current value is 100,  Default value is 100.
    Minimum value is 50,  Maximum value is 10000.

   The BASEstar Classic parameter ILAN$DTL_UPL_FLAGS has just
   been created.

Parameter ILAN$DTL_UPL_FLAGS, type String.
    Current value is "NORUNNING,NOFAULT",  Default value is "NORUNNING,NOFAULT".

   The BASEstar Classic parameter ILAN$DTL_DWNL_FLAGS has just
   been created.

Parameter ILAN$DTL_DWNL_FLAGS, type String.
    Current value is "MATCH",  Default value is "MATCH".

   The BASEstar Classic parameter ILAN$DTL_SCMP_FLAGS has just
   been created.

Parameter ILAN$DTL_SCMP_FLAGS, type String.
    Current value is "ALL,NODATATABLES,NOREVISION",  Default value is "ALL,NODAT
ATABLES,NOREVISION".

   The BASEstar Classic parameter ILAN$DTL_UDC_TIMEOUT has just
   been created.

Parameter ILAN$DTL_UDC_TIMEOUT, type Value.
    Current value is 300000,  Default value is 300000.
    Minimum value is 1000,  Maximum value is 3600000.

   The BASEstar Classic parameter ILAN$DTLSRV_PRIORITY has just
   been created.

Parameter ILAN$DTLSRV_PRIORITY, type Value.
    Current value is 6,  Default value is 6.
    Minimum value is 1,  Maximum value is 10.
```

```
   The BASEstar Classic parameter ILAN$DTL_DUP_EI_ID has just been created.

Parameter ILAN$DTL_DUP_EI_ID, type Bit.
    Current value is 0,  Default value is 0.
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...

   Copyright 1992 Compaq Computer Corporation

   Confidential commercial computer software.  Valid license required.

   Executing the Installation Verification Procedure.


Parameter ILAN$DTL_BACKLOG, type Value.
    Current value is 5,  Default value is 5.
    Minimum value is 1,  Maximum value is 40.

Parameter ILAN$DTL_CYCLE_TIME, type Value.
    Current value is 70,  Default value is 70.
    Minimum value is 5,  Maximum value is 1000.

Parameter ILAN$DTL_MAX_ELEMENTS, type Value.
    Current value is 500,  Default value is 500.
    Minimum value is 50,  Maximum value is 100000.

Parameter ILAN$DTL_MAX_UNSOL_ELEMENTS, type Value.
    Current value is 100,  Default value is 100.
    Minimum value is 50,  Maximum value is 10000.

Parameter ILAN$DTL_TIMEOUT, type Value.
    Current value is 1000,  Default value is 1000.
    Minimum value is 100,  Maximum value is 600000.

Parameter ILAN$DTL_UPL_FLAGS, type String.
    Current value is "NORUNNING,NOFAULT",  Default value is "NORUNNING,NOFAULT".

Parameter ILAN$DTL_DWNL_FLAGS, type String.
    Current value is "MATCH",  Default value is "MATCH".

Parameter ILAN$DTL_SCMP_FLAGS, type String.
    Current value is "ALL,NODATATABLES,NOREVISION",  Default value is "ALL,NODAT
ATABLES,NOREVISION".

Parameter ILAN$DTL_UDC_TIMEOUT, type Value.
    Current value is 300000,  Default value is 300000.
    Minimum value is 1000,  Maximum value is 3600000.

Parameter ILAN$DTLSRV_PRIORITY, type Value.
    Current value is 6,  Default value is 6.
    Minimum value is 1,  Maximum value is 10.

Parameter ILAN$DTL_DUP_EI_ID, type Bit.
    Current value is 0,  Default value is 0.

   BASEstar Classic DAS for Allen-Bradley INTERCHANGE Software installation
   procedure has succeeded.

        Installation of DCM_DTLVVA V3.4 completed at 14:20

        VMSINSTAL procedure done at 14:20
```

2.2.1 Files Created During Installation

Table 2–2 and Table 2–3 lists the files created by the DAS for AB INTERCHANGE software installation procedure, and the directories in which those files are placed for both the INTERCHANGE OpenVMS (Ethernet) and PI MicroVAX options.

Table 2–2 DAS Files Created (EI Network Interface Option)

Directory	Filename
BCC$SYSDATA:	DCM_DTL$MSG.DAT
	DCM_DTL_CONFIG.TEMPLATE
	DCM_DTL$SRV_STARTUP.COM
	DCM_DTL$SRV_QUOTAS.DAT
BCC$SYSTEM:	DCM_DTL$SRV.EXE
	DCM_DTL$EXPORT_6200.EXE
	DCM_DTL$IMPORT_6200.EXE
	DCM_DTL$SRV_STOP.EXE
SYS$HELP:	DCM_DTLVVA034.RELEASE_NOTES (VAX)
	DCM_DTLVAA034.RELEASE_NOTES (Alpha)
SYS$LIBRARY:	ILAN_DTL.EXE
	ILAN_DTLNET.EXE
	ILAN_DTL_SCMP.EXE (VAX)

Table 2–3 DAS Files Created (PI MicroVAX Option)

Directory	Filename
BCC$SYSDATA:	DCM_DTL$MSG.DAT
	DCM_DTL_CONFIG.TEMPLATE
	DCM_DTL$BPISRV_STARTUP.COM
	DCM_DTL$SRV_QUOTAS.DAT
BCC$SYSTEM:	DCM_DTL$BPISRV.EXE
	DCM_DTL$SRV_STOP.EXE
SYS$HELP:	DCM_DTLVVA034.RELEASE_NOTES (VAX)
	DCM_DTLVAA034.RELEASE_NOTES (Alpha)
SYS$LIBRARY:	ILAN_DTL.EXE
	ILAN_DTLBPI.EXE
SYS$MESSAGE:	DTL$MSG.EXE

Table 2–4 lists the files modified by the DAS for AB INTERCHANGE software installation procedure, and the directories in which those files are located.

Table 2–4 Files Modified

Directory	Filename
SYS$HELP:	ILAN$DEVICE_SPECIFIC.HLB

2.2.2 Installation Messages

You may see VMSINSTAL messages during the installation procedure. The following messages are specific to the DAS for AB INTERCHANGE software installation:

BADBCC, BASEstar Classic software must be installed before the DAS for AB INTERCHANGE Software,

Explanation: Error. BASEstar Classic software must be installed before installing the DAS for AB INTERCHANGE Software.

User Action: Install BASEstar Classic software, Version 3.4.

BADDCM, BASEstar Classic Device Connect software must be installed before the DAS for AB INTERCHANGE Software,

Explanation: Error. BASEstar Classic Device Connect software must be installed before installing the DAS for AB INTERCHANGE Software.

User Action: Install BASEstar Classic Device Connect Management software, Version 3.4.

BADVMS (Alpha), DAS for AB INTERCHANGE Software must be installed under OpenVMS V6.1.,

Explanation: Error. The DAS is being installed on a version of OpenVMS that is less than OpenVMS V6.1.

User Action: Upgrade OpenVMS to the required minimum version and then install the DAS.

BADVMS (VAX), DAS for AB INTERCHANGE Software must be installed under OpenVMS V5.5-2.,

Explanation: Error. The DAS is being installed on a version of OpenVMS that is less than OpenVMS V5.5-2.

User Action: Upgrade OpenVMS to the required minimum version and then install the DAS.

DTLBPI (VAX only), Installing MicroVAX version of the DAS...,

Explanation: Information. The INTERCHANGE Software for the PI MicroVAX version of the DAS for AB INTERCHANGE software has been selected to be installed..

User Action: None. Continue with installation.

DTLNET, Installing Ethernet version of the DAS...,

Explanation: Information. The INTERCHANGE Software for OpenVMS version of the DAS for AB INTERCHANGE software has been selected to be installed..

User Action: None. Continue with installation.

NETBLOCKS (VAX), The DAS for AB INTERCHANGE Software requires 500 blocks after installation,

Explanation: Error. There is not enough disk space available to install DAS for AB INTERCHANGE software.

User Action: Free up sufficient disk space to complete the installation.

NETBLOCKS (Alpha), The DAS for AB INTERCHANGE Software requires 1000 blocks after installation,

Explanation: Error. There is not enough disk space available to install DAS for AB INTERCHANGE software.

User Action: Free up sufficient disk space to complete the installation.

NODTLNET, Allen-Bradley INTERCHANGE software must be installed before the DAS for AB INTERCHANGE Software,

Explanation: Error. Correct version of INTERCHANGE Software for OpenVMS must be installed before the DAS for AB INTERCHANGE software.

User Action: Install INTERCHANGE Software for OpenVMS.

NODTLPI, Allen-Bradley INTERCHANGE software must be installed before the DAS for AB INTERCHANGE Software,

Explanation: Error. Correct version of INTERCHANGE Software for the PI MicroVAX must be installed before DAS for AB INTERCHANGE software.

User Action: Install INTERCHANGE Software for the PI MicroVAX.

NOLICENSE, No license found for this product - IVP will not be run.,

Explanation: Informational. A valid license was not found. The installation will continue, but the IVP will not be run.

User Action: Register and load a valid license for this product before attempting to use the DAS.

NOLOAD, License for this product not loaded - IVP will not be run.,

Explanation: Informational. The license for this product has not been loaded by the License Management Utility. The installation willl proceed, but the IVP will not be run.

User Action: Load the license using the License Management Utility before attempting to use the DAS.

2.3 Postinstallation Tasks

Before using the DAS for AB INTERCHANGE software, you must perform the following postinstallation tasks:

- Set TCP/IP for OpenVMS UCX parameters (for EI Network Interface option only).
- Edit the configuration file
- Set the BASEstar Classic port quota parameter
- Set the device connection management SPT (support) block parameter
- Set the DAS-specific BASEstar Classic parameters
- Set the DTL server process quotas

- Set up plant-floor equipment

2.3.1 Modify TCP/IP for OpenVMS UCX parameters

The Allen-Bradley INTERCHANGE Software for OpenVMS uses TCP/IP services for OpenVMS (UCX) to communicate to the Ethernet interface. By default, UCX waits 10 minutes before notifying a TCP/IP application that a connection has been broken or that a connection to an unavailable interface has failed. This default timeout characteristic may not be appropriate for your use with the DAS for AB INTERCHANGE software. To override this behavior, the following must be done.

In the system start up file SYS$MANAGER:SYSTARTUP_V5.COM, after SYS$STARTUP:UCX$STARTUP.COM, but before SYS$STARTUP:DTLNET_STARTUP.COM is invoked add the following two lines:

```
$UCX SET PROTOCOL TCP/PROBETIME=pt
$UCX SET PROTOCOL TCP/DROPTIME=dt
```

Where:

pt = number of seconds between sending keepalive probe messages. Default value is 75 seconds. Allen-Bradley recommended value is 10 seconds.
dt = number of seconds to wait since the last successful probe message before dropping the connection. The default value for *dt* is 600 seconds (10 minutes). The recommended value is 80 seconds. Note, the rule for setting *dt* is $dt \geq 8 \times pt$.

Consult the appropriate TCP/IP Services for OpenVMS and Allen-Bradley documentation for more detailed information on configuring the TCP/IP network for INTERCHANGE Software for OpenVMS use.

2.3.2 Editing the Configuration File

A configuration file, BCC$SYSDATA:DCM_DTL_CONFIG.TEMPLATE, is supplied with the DAS for AB INTERCHANGE software. This file contains definitions for types, paths, and devices. Copy the template file and edit it to include site-specific information about types, paths, and devices. Then execute the file to create the BASEstar Classic device connection management definitions.

The following sections provide examples of type, path, and device definitions from the configuration file. For more information about creating type, path, and device definitions, refer to the *BASEstar Classic Command Line Interface User's Guide*.

2.3.2.1 Editing Type Records

The type record associates a device with the correct protocol emulator (PE). Example 2–1 shows the configuration file commands that create type records.

Example 2–1 Type Configuration

```
create type DTL_PLC5250_TYPE -
                  /model="PLC-5/250"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/250"/log
```

(continued on next page)

Example 2–1 (Cont.) Type Configuration

```
create type DTL_PLC5_TYPE -
                  /model="PLC-5"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley generic PLC-5"/log

create type DTL_PLC5E_TYPE -
                  /model="PLC-5E"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley generic PLC-5, Ethernet interface"/log

create type DTL_PLC510_TYPE -
                  /model="PLC-5/10"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/10"/log

create type DTL_PLC512_TYPE -
                  /model="PLC-5/12"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/12"/log

create type DTL_PLC515_TYPE -
                  /model="PLC-5/15"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/15"/log

create type DTL_PLC525_TYPE -
                  /model="PLC-5/25"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/25"/log

create type DTL_PLC520E_TYPE -
                  /model="PLC-5/20E"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/20E"/log

create type DTL_PLC520C_TYPE -
                  /model="PLC-5/20C"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/20C"/log

create type DTL_PLC526E_TYPE -
                  /model="PLC-5/26E"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/26E"/log

create type DTL_PLC540_TYPE -
                  /model="PLC-5/40"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/40"/log
```

(continued on next page)

Example 2–1 (Cont.) Type Configuration

```
create type DTL_PLC560_TYPE -
                  /model="PLC-5/60"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/60"/log

create type DTL_PLC540L_TYPE -
                  /model="PLC-5/40L"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/40L"/log

create type DTL_PLC540E_TYPE -
                  /model="PLC-5/40E"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/40E"/log

create type DTL_PLC540C_TYPE -
                  /model="PLC-5/40C"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/40C"/log

create type DTL_PLC560L_TYPE -
                  /model="PLC-5/60L"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/60L"/log

create type DTL_PLC560C_TYPE -
                  /model="PLC-5/60C"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/60C"/log

create type DTL_PLC580E_TYPE -
                  /model="PLC-5/80E"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/80E"/log

create type DTL_PLC580C_TYPE -
                  /model="PLC-5/80C"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-5/80C"/log

create type DTL_PLC3_TYPE -
                  /model="PLC-3"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-3"/log

create type DTL_PLC2_TYPE -
                  /model="PLC-2"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-2"/log
```

(continued on next page)

Example 2–1 (Cont.) Type Configuration

```
create type DTL_PLC202_TYPE -
                  /model="PLC-2/02"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-2/02"/log

create type DTL_PLC205_TYPE -
                  /model="PLC-2/05"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-2/05"/log

create type DTL_PLC215_TYPE -
                  /model="PLC-2/15"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-2/15"/log

create type DTL_PLC216_TYPE -
                  /model="PLC-2/16"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-2/16"/log

create type DTL_PLC217_TYPE -
                  /model="PLC-2/17"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-2/17"/log

create type DTL_PLC220_TYPE -
                  /model="PLC-2/20"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-2/20"/log

create type DTL_PLC230_TYPE -
                  /model="PLC-2/30"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley PLC-2/30"/log

create type DTL_SLC500_TYPE -
                  /model="SLC-500"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley SLC-500"/log

create type DTL_SLC501_TYPE -
                  /model="SLC-5/01"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley SLC-5/01"/log

create type DTL_SLC502_TYPE -
                  /model="SLC-5/02"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley SLC-5/02"/log
```

(continued on next page)

Example 2–1 (Cont.) Type Configuration

```
create type DTL_SLC503_TYPE -
                  /model="SLC-5/03"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley SLC-5/03"/log

create type DTL_SLC504_TYPE -
                  /model="SLC-5/04"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley SLC-5/04"/log

create type DTL_SLC505_TYPE -
                  /model="SLC-5/05"-
                  /manufacturer="ALLEN-BRADLEY"-
                  /protocol=DTL -
                  /description="Allen-Bradley SLC-5/05"/log
```

Modify the command for your site configuration.

It should be noted that the Allen-Bradley model PLC-5/250 is **not** considered a member of the PLC-5 family. It is important that the model definition (from the TYPE) be consistent with that of the actual device. Attempts to perform functions on incorrectly defined devices will fail.

INTERCHANGE software does not distinguish between PLC-5 family processor models, therefore, only the DTL_PLC5_TYPE (for connecting off of the pyramid integrator) or DTL_PLC5E_TYPE (for direct connection to ethernet) need be used. The specific PLC-5 model types are provided as a convenience in configuration management of these device networks.

INTERCHANGE software also does not distinguish between PLC-3, PLC-2 or SLC family processor models, therefore, only the DTL_PLC3_TYPE, DTL_ PLC2_TYPE or DTL_SLC500_TYPE need be used. The specific PLC model types are provided as a convenience in configuration management of these device networks.

2.3.2.2 Editing Path Records

The two options for BASEstar Classic DAS for Allen-Bradley INTERCHANGE software have different network interfaces (NI), one to support each option of INTERCHANGE software. The path record associates a device with the correct NI and communication path.

Examples 2–2 and 2–3 show the configuration file commands that create records for Ethernet and PI paths.

Example 2–2 Path Configuration for EI Network Interface

```
create path DTLNET_PATH/vaxport="05.00.00.04"-
           /netname=DTLNET-
           /default/io_size=2000/multidrop/log
```

Example 2–3 Path Configuration for PI MicroVAX

```
create path DTLBPI_PATH/vaxport="NL:"/netname=DTLBPI-
           /default/io_size=2000/multidrop/log
```

Since the two options of the DAS for AB INTERCHANGE software are mutually exclusive, the */NETNAME* must correspond to the option that has been installed.

For the DAS for AB INTERCHANGE software ethernet option, the /NETNAME parameter must be DTLNET. The VAXPORT parameter of the path definition specifies the network address of the Ethernet interface module connecting the PLC with the TCP/IP network and optionally the DTL server to use when connecting. This address may be either a host name or an IP address specified in dot notation, with the same value that would be passed to the INTERCHANGE software routine DTL_C_CONNECT. Refer to Allen-Bradley documentation for more information.

To specify the DTL server process to use when connecting use the syntax:

```
/vaxport=xx.xx.xx.xx:yy
```

or

```
/vaxport=host_name:yy
```

where yy is the number of the DTL server process (from 1-64) to use when communicating with the Ethernet Interface. A maximum of 40 Ethernet Interfaces can be configured to use the same DTL server process.

For the PI MicroVAX option of the DAS for AB INTERCHANGE software, the /NETNAME parameter must be DTLBPI and the VAXPORT parameter must be NL:. Only one path may be defined when using this option.

Table 2–5 lists the path parameters and indicates the value(s) that are allowed for each.

Table 2–5 Path Parameters

Parameter	Value(s)
VAXport[1]	xx.xx.xx.xx, host_name xx.xx.xx.xx:yy, host_name:yy NL:
Netname	DTLNET DTLBPI
Multidrop	MULTIDROP
Timeout[2]	
Retries[3]	
IO Size[4]	

[1]See the above discussion for details on setting this parameter.

[2]The parameter is ignored by this DAS. The timeout for an I/O operation is controlled by the parameter ILAN$DTL_TIMEOUT.

[3]The parameter is ignored by this DAS. The DAS does not do retries.

[4]The parameter is ignored by this DAS.

The line parameters in the path definition are not used by this DAS and are ignored.

Modify the commands for your site configuration, making sure that the VAX port parameters match your plant-floor configuration. Create additional paths as needed.

2.3.2.3 Editing Device Records

The device record represents a Allen-Bradley device and references previously defined type and path records. Example 2–4 shows the configuration file commands that create device records.

Example 2–4 Device Configuration

```
create dev DTL5 /path=DTLNET_PATH/type=DTL_PLC5_TYPE/netaddr="0RM:2,03"-
              /timeout=100/vaxport="aa.00.00.bb"-
              /write_enable/log

create dev DTL5E /path=DTLNET_PATH/type=DTL_PLC5E_TYPE/netaddr=","-
              /timeout=100/vaxport="aa.00.00.bb"/write_enable/log

create dev DTL5250 /path=DTLNET_PATH/type=DTL_PLC5250_TYPE/netaddr=","-
              /timeout=100/vaxport="aa.00.00.bb"/write_enable/log

create dev SLC504 /path=DTLNET_PATH/type=DTL_SLC504_TYPE/netaddr="0RM:1,04"-
              /timeout=100/vaxport="aa.00.00.bb"/write_enable/log

create dev PI5250 /path=DTLBPI_PATH/type=DTL_PLC5250_TYPE/netaddr=","-
              /timeout=100/vaxport="NL:"/write_enable/log
```

The timeout on the device definition controls the time that BASEstar Classic device connection management allows for a device operation to complete. The value for the device timeout should be larger than than the expected time of the longest device operation and also longer than the timeout used by INTERCHANGE software. To control the timeout that INTERCHANGE software uses to timeout requests, use the ILAN$DTL_TIMEOUT parameter. See Section 2.3.5.8 for information on configuring this parameter.

For the DAS for AB INTERCHANGE software, the /NETADDRESS parameter specifies the external communication port and station address of a station outside of the Pyramid integrator chassis respectively. The format of the /NETADDRESS parameter is: *port_id*,*station_id*.

Valid *port_id* values are as follows:

Port ID	Description
2 or 0RM:2	Channel 2 of the resource manager in slot 0
3 or 0RM:3	Channel 3 of the resource manager in slot 0
nKA:2	Channel 2 of the KA module with pushwheel n(n=0-4)
nKA:3	Channel 3 of the KA module with pushwheel n(n=0-4)
0RM:2/B:b	Channel 2 of the resource manager through bridge address b (b=1-376(octal))
0RM:2/B:b/L:l	Channel 2 of the resource manager through bridge address b (b=1-376(octal)) and link l (l=0-65535(decimal))
0RM:2/L:l/G:g	Channel 2 of the resource manager through link l (l-0-65535(decimal)) and gateway g (g=1-376(octal))
0RM:2/B:b/L:l /G:g	Channel 2 of the resource manager through bridge address b (b=1-376(octal)) link l (l=0-65535(decimal)) and gateway g (g=1-376(octal))

Valid value for the *station_id* are 1 through 376 (octal) for Data Highway, and 0 through 77 (octal) for Data Highway Plus.

For example, a network address at Data Highway station 7 off of the DH/DH+ port 2 of the resource manager is as follows: `/NETADDRESS="0RM:2,07"`

For a PLC-5/250 sharing the Pyramid integrator chassis with the Ethernet Interface module, the *port_id* and *station_id* are null or empty.

For PLC-5/XXE or SLC-5/05 with a direct connection to the ethernet, the *port_id* and *station_id* are also null.

To specify the network address for either of these configurations, use the comma (,) delimiter, as shown: `/NETADDRESS=","`

Modify the command for your site configuration.

For more information about maintaining type, path, and device definitions, refer to the *BASEstar Classic Command Line Interface User's Guide*.

2.3.3 Setting the BASEstar Classic Message Port Quota Parameter

The DAS for AB INTERCHANGE software requires a message port quota of 100,000. The BASEstar Classic parameter BCC$MAX_PORT_QUOTA must be set greater than or equal to this value.

Use the BASEstar Classic CLI SHOW PARAMETER command to review the value of the maximum port quota parameter. If necessary, use the SET PARAMETER command to adjust it. For more information refer to the *BASEstar Classic Configuration and Tuning Guide*.

2.3.4 Setting the SPT Block Parameter

The ILAN$MAX_SPT_REQUESTS parameter specifies the total number of blocks that can be allocated in the SPT (support) global section. DASes use blocks in the global section for storing data structures and for doing device I/O. The SPT global section is sized by calculating the number of SMALL, MEDIUM, LARGE and EXTRA LARGE blocks that the section should contain. Some blocks remain for the life of a device and some are allocated and deallocated for each I/O operation. Table 2–6 shows the static blocks of each size that are used by the DAS.

Table 2–6 SPT Static Block Sizes

Block Size	Quantity[1]	Block Type
MEDIUM	2	Device
	1	Line
	1	DTL Server
	1	DAS
SMALL	1	Device[2]
	1	Unsolicited point[2]

[1]Quantity is quantity per device, per line, per server, etc.

[2]Only created if the device is marked "unsolicited".

Table 2–7 shows the dynamic blocks of each size that are used by the DAS. These blocks are created and deleted as the device does I/O.

Table 2–7 SPT Dynamic Block Sizes

Block Size	Quantity[1]	I/O Type
LARGE	1	Status
MEDIUM	1	Read[2], write[2], upload, download
SMALL	3	Any[3]

[1]Quantity is quantity per I/O.

[2]Block size is dependent on the size of I/O. For data larger than ~345 bytes, block size will be LARGE.

[3]These blocks are created for any I/O operation.

The size of the SPT global section can be tuned by changing the percentage of each kind of block that is created. Refer to the *BASEstar Classic Configuration and Tuning Guide* for instructions on changing the percentage of each size of block that is created in the global section.

2.3.5 Setting DAS-Specific Parameters

The following sections describe how to set the following DAS-specific BASEstar Classic parameters:

- ILAN$DTLSRV_PRIORITY
- ILAN$DTL_BACKLOG
- ILAN$DTL_CYCLE_TIME
- ILAN$DTL_DWNL_FLAGS
- ILAN$DTL_MAX_ELEMENTS
- ILAN$DTL_MAX_UNSOL_ELEMENTS
- ILAN$DTL_SCMP_FLAGS
- ILAN$DTL_TIMEOUT
- ILAN$DTL_UDC_TIMEOUT
- ILAN$DTL_UPL_FLAGS
- ILAN$DTL_DUP_EI_ID

None of the parameters created for the DAS for AB INTERCHANGE software are dynamic. BASEstar Classic device connection management must be shut down and started again before parameter modifications become effective.

2.3.5.1 ILAN$DTLSRV_PRIORITY

The ILAN$DTLSRV_PRIORITY parameter sets the priority of the INTERCHANGE server process, DCM_DTL$SRV. Values are as follows:

Default value	6
Minimum value	1
Maximum value	10

Use this parameter in conjunction with the ILAN$DTL_CYCLE_TIME parameter to balance the performance of the INTERCHANGE server process. A high priority setting will give more CPU time to the server process for collecting data and processing requests. Understand, however, that increasing this setting may be at the expense of the applications using this data. Refer to the appropriate documentation on OpenVMS system management and tuning for more information.

2.3.5.2 ILAN$DTL_BACKLOG

The ILAN$DTL_BACKLOG parameter sets the number of unsolicited messages which will be buffered by the INTERCHANGE software. Values are as follows:

Default value	5 messages
Minimum value	1 message
Maximum value	40 messages

2.3.5.3 ILAN$DTL_CYCLE_TIME

The ILAN$DTL_CYCLE_TIME parameter controls the maximum amount of time the INTERCHANGE server process waits for DTL I/O completion before checking for requests from the NI. Values are as follows:

Default value	70 milliseconds
Minimum value	5 milliseconds
Maximum value	1,000 milliseconds

Generally, the smaller the cycle time, the faster the server will respond to the NI requests. If the cycle time is too small, however, the server will consume CPU time checking for requests.

2.3.5.4 ILAN$DTL_DWNL_FLAGS

The ILAN$DTL_DWNL_FLAGS parameter sets the default behavior of the download function. This default behavior may be overridden by using /QUALIFIER option on the download command. The flags specified in a download command are logically ORed with those specified by ILAN$DTL_DWNL_FLAGS.

The default value for ILAN$DTL_DWNL_FLAGS is MATCH. This flag requires that the device type, series, and revision match that of the file being downloaded. For more information on valid download flags that can be used for this parameter, refer to Table 3–8.

2.3.5.5 ILAN$DTL_MAX_ELEMENTS

The ILAN$DTL_MAX_ELEMENTS parameter controls the maximum number of elements the INTERCHANGE server process can define; that is, the expected number of outstanding solicited requests and the number of unsolicited requests combined. Values are as follows:

Default value	500 elements
Minimum value	50 elements
Maximum value	100,000 elements

The value of ILAN$DTL_MAX_ELEMENTS must be greater than the value of ILAN$DTL_MAX_UNSOL_ELEMENTS. Note that ILAN$DTL_MAX_ ELEMENTS only affects the number of points that can be defined as unsolicited and simultaneous pending data reads/writes. The parameter does not affect the number of physical points that can be defined on a BASEstar Classic system.

2.3.5.6 ILAN$DTL_MAX_UNSOL_ELEMENTS

The ILAN$DTL_MAX_UNSOL_ELEMENTS parameter controls the maximum number of elements that can be defined as unsolicited phypoints. This number is deducted from the value of ILAN$DTL_MAX_ELEMENTS. Values are as follows:

Default value	50 phypoints
Minimum value	50 phypoints
Maximum value	10,000 phypoints

The value of ILAN$DTL_MAX_UNSOL_ELEMENTS must be less than ILAN$DTL_MAX_ELEMENTS.

2.3.5.7 ILAN$DTL_SCMP_FLAGS

The ILAN$DTL_SCMP_FLAGS parameter sets the default behavior of the simple file compare function.

The default
value for ILAN$DTL_SCMP_FLAGS is ALL,NODATATABLES,NOREVISION. This value will have the compare operation check everything except the data tables and the PLC revision. For more information on valid simple file compare flags which may be used for this parameter, refer to Table 3–12.

2.3.5.8 ILAN$DTL_TIMEOUT

The ILAN$DTL_TIMEOUT parameter determines the timeout value for all DTL I/O calls. Values are as follows:

Default value	1,000 milliseconds
Minimum value	1,000 milliseconds
Maximum value	600,000 milliseconds

The value of ILAN$DTL_TIMEOUT should be less than the shortest /TIMEOUT of the INTERCHANGE DAS device definitions. The desired behavior is to have the internal INTERCHANGE DTL call timeout before the BASEstar Classic device connect management request. If the BASEstar Classic request times out first, a cancellation request is attempted and any requests completion is ignored. Because INTERCHANGE does not support cancellation of requests, the actual INTERCHANGE DTL request may have successfully completed, yet give the incorrect impression that the INTERCHANGE function was not completed.

2.3.5.9 ILAN$DTL_UDC_TIMEOUT

The ILAN$DTL_UDC_TIMEOUT parameter determines the timeout value for all INTERCHANGE upload, download and compare calls. Values are as follows:

Default value	300000 milliseconds
Minimum value	1000 milliseconds
Maximum value	3600000 milliseconds

The value of ILAN$DTL_UDC_TIMEOUT should be less than the shortest /TIMEOUT of the INTERCHANGE DAS device definitions. The desired behavior is to have the internal INTERCHANGE DTL call timeout before the BASEstar Classic device connect management request. If the BASEstar Classic request times out first, a cancellation request is attempted and any requests completion is ignored. Because INTERCHANGE does not support cancellation of requests, the actual INTERCHANGE DTL request may have successfully completed, yet give the incorrect impression that the INTERCHANGE function was not completed.

2.3.5.10 ILAN$DTL_UPL_FLAGS

The ILAN$DTL_UPL_FLAGS parameter sets the default behavior of the upload function. This default behavior may be overridden by using /QUALIFIER option on the upload command. The flags specified in a upload command are logically ORed with those specified by ILAN$DTL_UPL_FLAGS.

The default value for ILAN$DTL_UPL_FLAGS is NORUNNING,NOFAULT, which requires that the device not be in RUN mode and that it not be faulted. For more information on valid upload flags which may be used for this parameter, refer to Table 3–9.

2.3.5.11 ILAN$DTL_DUP_EI_ID

The ILAN$DTL_DUP_EI_ID parameter allows the DAS to communicate to more than forty (40) Ethernet Interfaces. The maximum number of Ethernet Interface IDs is forty (40). This maximum is a limit imposed by INTERCHANGE software. The DAS will by default allocate a new Ethernet Interface ID whenever it communicates to a new Ethernet Interface. However, if this parameter is set to "1", then the DAS wil allocate up to forty (40) Ethernet Interfaces IDs per DTL server process, rather than using a total of forty (40) IDs for all DTL server processes.

If this parameter is set, then the DAS will create up to sixty-four (64) DTL server processes, with each process using Ethernet Interface IDs from 1-40. The number of DTL server processes can be tuned by explicitly specifying the server process to use when communicating to a particular Ethernet Interface. See Section 2.3.2.2 for details on configuring a path to explicitly specify the DTL server to use when communicating.

2.3.6 Setting the DTL Server Process Quotas

The DAS for AB INTERCHANGE software allows you to tune the values of process quotas for the DCM_DTL$SRV_X process. Process parameters are read from the file BCC$SYSDATA:DCM_DTL$SRV_QUOTAS.DAT when a new DCM_DTL$SRV process is created.

Process quotas may need to be tuned depending on the number of devices and points configured and the throughput rate of requests through the server. To adjust a quota for the server process do the following:

1. Provide DETACH priviledge to ILAN$DEVSRV by modifying the RUN command in BCC$SYSDATA:ILAN$SYSTEM_STARTUP.COM.
2. Edit the file BCC$SYSDATA:DCM_DTL$SRV_QUOTAS.DAT as follows:
    1. Un-comment the line corresponding to the quota to be adjusted by removing the "!" character.
    2. Modify the existing quota value. If a value does not exist, add it to the file, separating the quota name and value with a space.

The new quotas will be in effect the next time the server process is started up by the DAS for AB INTERCHANGE software. The DCM_DTL$SRV_STOP utility can be used to shutdown running server processes without shutting down BASEstar Classic DCM.

2.3.7 Setting Up Plant-Floor Equipment

To set up your plant-floor equipment, refer to the Allen-Bradley documentation for your specific device.

While setting up the plant-floor equipment, you should record the following information that will be required for setting up the BASEstar Classic path and device definitions:

- The Ethernet Address of the Ethernet Interface(s)
- The Data Highway addresses of remote PLC stations

2.4 Tracing Device Communications

The DAS for AB INTERCHANGE software has built into it the capability to trace INTERCHANGE software communications and to send this output to either a file, a user terminal or to the BASEstar Classic history file. The purpose of this tracing is to allow the troubleshooting of INTERCHANGE software communications and is intended to be used primarily under the direction of support personnel.

To enable tracing to send trace information to a file do the following:

- Set the following logical to 1:

  DCM_DTL$SRV_TRACE Trace the DCM_DTL$SRV process(es)

  The logical name must be defined in a scope that will be seen by the associated server process (DCM_DTL$SRV_n).

- Set the corresponding output logical to a file or other output device:

  DCM_DTL$SRV_TRACE_OUTPUT_n Trace the DCM_DTL$SRV process "n"

  The logical name must be defined in a scope that will be seen by the associated server process (DCM_DTL$SRV_n).

- Start the tracing.

  To start the tracing for the DCM_DTL$SRV process, stop the associated server process and restart it. The trace logicals are read only at process startup.

The trace output is flushed to the file after every 100 lines that are traced, so trace output may lag actual device I/O.

To turn tracing off, reverse the steps used to turn tracing on.

2.5 Failures During Product Use

If an error occurs while this product is in use and you believe the error is caused by a problem with the product, take one of the following actions:

- If you have a Software Product Services Support Agreement, contact your Customer Support Center (CSC) by telephone or by using the electronic means provided with your support agreement (such as DSNlink). The CSC provides telephone support for high-level advisory and remedial assistance. When you initially contact the CSC, indicate the following:
  - The name and version number of the operating system you are using
  - The version number of the product you are using
  - The version number of INTERCHANGE software you are using
  - The version number of BASEstar Classic you are using
  - The hardware system you are using (such as a model number)
  - The Allen-Bradley PLCs you are communicating with
  - A brief description of the problem (one sentence if possible)
  - How critical the problem is
- If you have a Self-Maintenance Software Agreement, you can submit a Software Performance Report (SPR).

- If you do not have any type of software services support agreement and you purchased this product within the past year, you can submit an SPR if you think the problem is caused by a software error.

When you submit an SPR, take the following steps:

1. Describe as accurately as possible the circumstances and state of the system when the problem occurred. Include the description and version number of the product being used. Demonstrate the problem with specific examples.
2. Reduce the problem to as small a size as possible.
3. Remember to include listings of any command files, INCLUDE files, or relevant data files, and so forth.
4. Report only one problem per SPR. This will facilitate a faster response.
5. Mail the SPR package to Compaq.

3
Using the DAS

This chapter provides information about the supported functions for Allen-Bradley devices, and how to access these functions.

3.1 Accessing DAS Functions

DAS for AB INTERCHANGE software functions are accessed through the following BASEstar Classic device connection management interfaces:

- Command line interface (CLI)
- Menu interface
- Application programming interface (API)

Command Line Interface (CLI) To access the BASEstar Classic device connection management CLI, enter the following command at the DCL prompt ($):

```
$ BSTAR DCM
```

For information about BASEstar Classic device connection management commands, refer to the *BASEstar Classic Command Line Interface User's Guide*.

Menu Interface To access the BASEstar Classic device connection management menu interface, enter the following command:

```
$ BSTAR/MENU
```

For additional information about the BASEstar Classic device connection management menu interface, refer to the *BASEstar Classic Menu Interface User's Guide*.

Application Programming Interface (API) For information on accessing BASEstar Classic device connection management callable services, refer to the *BASEstar Classic Introduction to Callable Services*.

Not all device connection management functions are available through every interface. When a function is limited to a particular interface, it is described specifically in the description of the function. If a limitation is not specified, the function is available through all device connection management interfaces.

3.2 Supported Functions

Table 3–1 describes the functions supported by the DAS for AB INTERCHANGE software.

Table 3–1 Supported Devices and Functions

	Devices				
Function	**PLC 5/250**	**PLC-5 Family**	**PLC-3 Family**	**PLC-2 Family**	**SLC Family**
Read Data	X	X	X	X	X
Write Data	X	X	X	X	X
Start					
Stop					
Upload		X[1]			
Download		X[1]			
Read Status	X	X	X	X	X
Loop	X	X	X	X	X
Delete File					
Select File					
Directory					
Allocate—Deallocate					
Device-Specific					
Simple Compare		X[1]			

[1]Valid only for the EI Network Interface option.

3.2.1 Read and Write Data Functions

Use the BASEstar Classic device connection management read data and write data functions to read data from and write data to a specific address in device memory. Address syntax is specific to each device or device family.

3.2.1.1 PLC-5/250 and PLC-5 Addressing

The general format for specifying addresses for the PLC-5/250 family devices is:

```
$mFSf:nn/bb
```

or

```
$mFSf:rr/bb
```

or

```
$mFSf:nn.mne
```

The general format for specifying addresses for the PLC-5 family devices is:

```
$FSf:nn/bb
```

or

```
$FSf:rr/bb
```

or

```
$FSf:nn.mne
```

where:

m—module number - used for the PLC-5/250 only.
FS—file specifier - specifies the device file type
f —decimal file number
nn—decimal element number
rr—octal rack number
bb—bit address - octal if following rr, decimal if following nn
mne—sub-element or bit mnemonic

The file specifiers for the PLC-5/250 and the PLC-5 family that are supported by the DAS for AB INTERCHANGE software are listed in Table 3–2.

Table 3–2 PLC-5 Supported Section Files

Section	Mnemonic	Native data type	Raw Size in bytes	
			PLC-5/250	PLC-5 family
System Status	S	Signed word	2	2
Binary	B	Signed word	2	2
Integer	N	Signed word	2	2
ASCII	A	Signed word	—	2
BCD	D	BCD	—	2
Long	L	Signed Long	4	—
Floating Point	F	IEEE single precision	4	4
Timer	T	Structure	12	6
Counter	C	Structure	6	6
Message	MSG	Structure	112	—
Message	MG	Structure	—	112[1]
Control	R	Structure	6	6
PID	PD	Structure	164	164[1]
String	ST	Structure	84	84[1]
Input Image	I	Signed word	2	2
Output Image	O	Signed word	2	2

[1]Not available for PLC-5 models 10, 12, 15 and 25.

For specific information regarding the addressing syntax for the Allen-Bradley PLC-5/250 and the PLC-5 family, refer to Allen-Bradley documentation for those devices.

3.2.1.2 SLC Addressing

The general format for specifying addresses for the SLC family devices is:

```
$FSf:nn/bb
```

or

```
$FSf:nn.mne
```

where:

FS—file specifier - specifies the device file type
f —decimal file number
nn—decimal element number
bb—decimal bit address
mne—sub-element or bit mnemonic

The file specifiers for the SLC family that are supported by the DAS for AB INTERCHANGE software are listed in Table 3–3.

Table 3–3 SLC Supported Section Files

Section	Mnemonic	Native data type	Raw Size in bytes
System Status	S	Signed word	2
Binary	B	Signed word	2
Integer	N	Signed word	2
ASCII	A	Signed word	2
BCD	D	BCD	2
Floating Point	F	IEEE single precision	4
Timer	T	Structure	6
Counter	C	Structure	6
Control	R	Structure	6
String	ST	Structure	84
Input Image	I	Signed word	2
Output Image	O	Signed word	2

For specific information regarding the addressing syntax for the Allen-Bradley SLC family, refer to Allen-Bradley documentation for those devices.

3.2.1.3 PLC-3 Addressing

The INTERCHANGE software supports extended addressing only for the PLC-3 family of devices. The general format for extended addressing is:

```
EM.S.S.S.S.W/B
```

where:

E—extended address indicator
M—major area number
S—subarea designation
W—word address
B—bit address

A major area of 3 is used for data table addressing. For data table addressing, the first subarea is the context (1-15). The second subarea is the section, the third subarea is the file number, the fourth subarea is the structure number (for timers, counters, floating point, high order integer and pointers) and the fifth subarea is the word address (for output, input, integers, decimal, binary, ASCII, and status). Table Table 3–4 shows data table sections supported for the PLC-3 family of PLCs.

Table 3–4 PLC-3 Supported Section Files

Section	Subarea	Native data type	Raw Size in bytes
Output	1	Signed word	2
Input	2	Signed word	2
Timer	3	Structure	6
Counter	4	Structure	6
Integer	5	Signed word	2
Floating	6	VAX float	4
Decimal	7	Signed word	2
Binary	8	Signed word	2
ASCII	9	Signed word	2
High order integer	10	Signed long	4
Pointers	12	Signed long	4
Status	13	Signed word	2

For specific information regarding the addressing syntax for the Allen-Bradley PLC-3 family, refer to Allen-Bradley documentation for those devices.

3.2.1.4 PLC-2 Addressing

The INTERCHANGE software supports the following addressing format for PLC-2 family devices:

```
W/B
```

where:

W—word number (octal)
B—bit address (octal)

3.2.1.5 Reading and Writing Data

Table 3–5 describes how to access the read and write data functions through the BASEstar Classic interfaces.

Table 3–5 Read and Write Data Interface Access

Interface	Function Name
CLI	READ DATA
	READ PHYPOINT
	WRITE DATA
	WRITE PHYPOINT
API	ILAN$READ_DATA
	ILAN$READ_DATA_A
	ILAN$GET_DATA
	ILAN$WRITE_DATA
	ILAN$WRITE_DATA_A

(continued on next page)

Table 3–5 (Cont.) Read and Write Data Interface Access

Interface	Function Name
	ILAN$PUT_DATA
Menu	DCM Devices Read Register
	DCM Devices Read Phypoint
	DCM Devices Write Register
	DCM Devices Write Phypoint

Example 3–1 shows the information displayed when you enter the READ DATA command.

Example 3–1 READ DATA Output

```
DCM> READ DATA DTL5/FORMAT=WORD/ADDRESS="$N17:0"/NUMBER=20

Device      : DTL5
Address     : $N17:0
Format      : ARRAY[20]:S_WORD

Data:
    0       :        5        3        7      495     3504      495      366
    7       :      297     6252      222     8771     5132      495     5133
   14       :      495     5135      495     5139      495        0
```

Example 3–2 shows the information displayed when you enter the WRITE DATA command.

Example 3–2 WRITE DATA Output

```
DCM> WRITE DATA DTL5/FORMAT=BIT/ADDRESS="$B3:1/2"

Device      : DTL5
Address     : $B3:1/2
Format      : BIT

Data value     0       :        1
```

3.2.2 Read Status Function

The read status function issues a diagnostic status request to the device, interprets the device response, and returns the response as a character buffer. If a device definition does not match the device in the device response, an error message is displayed and the device definition must be corrected before you can perform an upload function or download function.

Table 3–6 shows how to access the read status function through the BASEstar Classic interfaces.

Table 3–6 Read Status Interface Access

Interface	Function Name
CLI	READ STATUS
API	ILAN$READ_DEVICE_STATUS
	ILAN$READ_DEVICE_STATUS_A
Menu	DCM Devices Status

For more information about the READ STATUS command, refer to the *BASEstar Classic Command Line Interface User's Guide*. Example 3–3, Example 3–4, Example 3–5, Example 3–6 and Example 3–7 provide example output of the READ STATUS command.

Example 3–3 READ STATUS Output for PLC-5/250

```
DCM> READ STATUS DTL_5250/FULL

 Status of device DTL_5250  at 22-DEC-1999 10:01:10.18

Server: DCM_DTL$SRV_1
Ethernet interface ID: 1
PLC Modes: Run
PLC Type: PLC-5/250       Communication adapters: 5130-RM
Octal station number: 0   Program change sequence: 54
Data change sequence: 18  User data change Sequence: 0
```

Example 3–4 READ STATUS Output for PLC-5

```
DCM> READ STATUS DTL_540/FULL

 Status of device  DTL_540  at 22-DEC-1999 10:03:02.55

Server: DCM_DTL$SRV_1
Ethernet interface ID: 1
PLC Modes: Rem. program
PLC Type: 1785-L40B (PLC-5/40B) Series B Revision B
Communication adapters: 1785-LT (PLC-5)  Octal station number: 3
Memory size in bytes: 98304
Number data table files: 40     Number program files: 5
I/O address if an adapter: 253   Baud rate:
```

Example 3–5 READ STATUS Output for SLC

```
DCM> READ STATUS DTL_SLC/FULL

 Status of device  DTL_SLC  at 22-DEC-1999 10:03:20.44

Server: DCM_DTL$SRV_1
Ethernet interface ID: 1
PLC Modes: Rem. program
PLC TYPES: 1747-L541 Series B Revision D
Communication adapters: Domino Plug
SLC500 Ascii Bulletin #: 5/04
SLC500 Major Error Code: 0
SLC500 Comms Active
SLC500 RAM Size in Kbytes: 16
```

Example 3–6 READ STATUS Output for PLC-3

```
DCM> READ STATUS DTL_3/FULL

 Status of device  DTL_3  at 22-DEC-1999 10:01:01.08

Server: DCM_DTL$SRV_1
Ethernet interface ID: 1
PLC Modes: Program
PLC TYPES: PLC-3 Series A Revision E
Communication adapters: 1775-KA DH port
Current context: 1; Thumbwheel number: 1
Mode control word: 35096
PLC-3 Physical Address of E60.0.0.0: 3574
PLC-3 total number of words in memory: 32768
```

Example 3–7 READ STATUS Output for PLC-2

```
DCM> READ STATUS DTL_2/FULL

 Status of device  DTL_2  at 22-DEC-1999 10:06:29.33

Server: DCM_DTL$SRV_1
Ethernet interface ID: 1
PLC Modes: Program
PLC TYPES: PLC-2/30 Series A Revision F
Communication adapters: 1771-KA(2)
```

The read status function allows you to verify if the BASEstar Classic device definition MODEL attribute is compatible with the corresponding physical device. If the BASEstar Classic definition is not compatible, the read status function displays an informational message stating what the actual device is and what it was defined as in BASEstar Classic.

3.2.3 Upload and Download Functions

The upload function transfers the contents of a device's memory to an OpenVMS file. The download function transfers the contents of an OpenVMS file to a device's memory. Upload and download functions are only supported for the EI Network Interface version of the DAS. Only PLC-5 processors (not including the PLC-5/250) may be uploaded or downloaded.

Table 3–7 describes how to access the upload and download functions through the BASEstar Classic interfaces.

Table 3–7 Upload and Download Interface Access

Interface	Function Name
CLI	UPLOAD
	DOWNLOAD
API	ILAN$UPLOAD_FILE
	ILAN$UPLOAD_FILE_A
	ILAN$DOWNLOAD_FILE
	ILAN$DOWNLOAD_FILE_A
Menu	DCM Devices Files Upload
	DCM Devices Files Download

Example 3–8 shows the CLI commands to upload and download files.

Example 3–8 UPLOAD and DOWNLOAD Input

```
DCM> UPLOAD device_name filename /QUALIFIER="RUNNING"
DCM> DOWNLOAD device_name filename /QUALIFIER="CLEAR"
```

In the first command, the BASEstar Classic device *device_name* is being uploaded to file *filename*. The "RUNNING" qualifier allows the PLC to be uploaded while it is in RUN or REMOTE RUN mode. In the second command, the file *filename* is to be downloaded to *device_name*. The CLEAR qualifier is used to clear the faulted PLC, allowing the download to proceed.

The DAS for AB INTERCHANGE software uses the Allen-Bradley INTERCHANGE Software for OpenVMS DTL_DOWNLOAD and DTL_ UPLOAD services to perform these functions. Consequently, the DAS for AB INTERCHANGE software upload and download functions have the same functional limitations.

Note that communication failures between the BASEstar Classic system and a device being uploaded or downloaded may leave a device in an state that prevents further access from the DAS for AB INTERCHANGE software. If such an error occurs, refer to the appropriate Allen-Bradley documentation for that device for trouble shooting assistance.

3.2.3.1 DTL_DOWNLOAD

For downloading, the OpenVMS file must have been created from the BASEstar Classic upload function or imported using the DCM_DTL$IMPORT_6200 utility. DCM_DTL$IMPORT_6200 creates a BASEstar Classic downloadable file from files generated from an Allen-Bradley's 6200 PLC-5 Programming Software environment or a INTERCHANGE Software for OpenVMS utility program.

The default behavior of the DAS for AB INTERCHANGE software download allows download providing all of the following conditions are met:

- The PLC is in program or remote program mode.
- There are no major faults.

- The revision and series of the PLC match the memory image to be downloaded.

Table 3–8 describes the qualifiers used to modify the default download behavior.

Table 3–8 Download Qualifiers

Qualifier	Default	Description
[NO]REMOTE	NOREMOTE	Enables download if the PLC is in REMOTE PROGRAM mode. The default is that the PLC must be in program mode.
[NO]CHANGE_MODE	NOCHANGE_MODE	Changes the mode of PLC from remote run or remote test to remote program before starting download. The default behavior does not change the mode at the beginning of the download, nor does it reset the mode to remote run or remote test after completing the download. The REMOTE qualifier must also be used with CHANGE_MODE to enable downloading in REMOTE PROGRAM mode.
[NO]CLEAR	NOCLEAR	Clears major faults on download if the PLC is faulted. The default is not to clear major faults.
[NO]MATCH	MATCH	Download only if the PLC type, revision and series are an exact match. The default is not to download if the revision and series do not match.

The default values of these qualifiers may be modified through setting the BASEstar Classic parameter ILAN$_DTL_DWNL_FLAGS.

Warning

Devices can control complex and perhaps dangerous industrial processes. Do not use the /CHANGE_MODE qualifier until you take the required safety precautions and put required operating restrictions into effect. Refer to the device manufacturer's documentation for specific safety precautions.

3.2.3.2 DTL_UPLOAD

The default behavior of the upload function for DAS for AB INTERCHANGE software is to allow an upload to occur if the target PLC is not running and is not faulted. This behavior may be modified by using the upload qualifiers described in Table 3–9.

Table 3–9 Upload Qualifiers

Qualifier	Default	Description
[NO]RUNNING	NORUNNING	Enables upload if the processor is in run mode. Note that the data table values may be changing.
[NO]FAULT	NOFAULT	Enables upload when PLC has a major fault. Note archive files created from faulted PLC may be downloaded but not compared.

To modify the default values of these qualifiers, use the BASEstar Classic parameter ILAN$_DTL_UPL_FLAGS.

3.2.4 Loop Diagnostic Function

The loop function sends a diagnostic loop-back message and waits for a response. The source of the response depends upon the extent of the diagnostic message as determined by the LOOP command qualifier or the loop_flag argument to the device connection management callable service. Table 3–10 describes the path of the diagnostic message according to the specified parameter.

Parameter	Response
/PE ILAN$K_LOOP_PE	Elapsed time
/NI ILAN$K_LOOP_NI	Elapsed time

The Loop PE function sends a message to the PE and a response is returned by the PE.

The Loop NI function is routed through the NI to the DTL server process. The server process returns the message back through the NI. The Loop NI function is a good method of determining the availability of the DTL server process.

Table 3–10 describes how to access the loop diagnostic function through the BASEstar Classic interfaces.

Table 3–10 Loop Interface Access

Interface	Function Name
CLI	LOOP
API	ILAN$LOOP_DEVICE
	ILAN$LOOP_DEVICE_A
Menu	Not available

Example 3–9 shows the information displayed by the LOOP command.

Example 3–9 Loop Output

```
DCM> LOOP DTL5/PE
 ELAPSED:    0 00:00:00.04  CPU: 0:00:00.03  BUFIO: 0  DIRIO: 0  FAULTS: 0

DCM> LOOP DTL5/NI
 ELAPSED:    0 00:00:00.11  CPU: 0:00:00.02  BUFIO: 0  DIRIO: 0  FAULTS: 0
```

3.2.5 Simple Compare Function

The simple compare function allows a BASEstar Classic library file related to a device to be compared to another existing BASEstar Classic library file of the same type. The BASEstar Classic library file must have been uploaded into the library directly or imported by the appropriate library function.

Simple compare is only supported by the EI Network Interface option of the DAS. Only PLC-5 family memory images (not including the PLC-5/250) may be compared.

Table 3–11 describes how to access the simple compare function through the BASEstar Classic interfaces.

Table 3–11 Simple Compare Interface Access

Interface	Function Name
CLI	Not available
API	Not available
Menu	Library File Compare File
Menu	Library External Compare Device

The default compare behavior for DAS for AB INTERCHANGE software is controlled by the BASEstar Classic parameter ILAN$DTL_SCMP_FLAGS. The values and defaults for these compare qualifiers are described in Table 3–12.

Table 3–12 Simple Compare Qualifiers

Qualifier	Default	Description
ALL		All compare options are checked
[NO]TYPE	TYPE	Memory images must be of the same PLC type
[NO]REVISION	NOREVISION	Memory images must be of the same PLC type, series and revision
[NO]MEMSIZE	MEMSIZE	Memory images must have the same memory size
[NO]NUMTABLES	NUMTABLES	Memory images must have the same number of data table files.
[NO]NUMPROGRAMS	NUMPROGRAMS	Memory images must have the same number of program files.
[NO]DATATABLES	NODATATABLES	Compare the data tables
[NO]FORCES	FORCES	Compare the forces
[NO]DIRECTORY	DIRECTORY	Compare program directories
[NO]LADDER	LADDER	Compare the ladder logic.

3.3 Supported Data Types

The following data types are supported by the DAS for AB INTERCHANGE software:

- BIT
- ARRAY[ ]:BIT
- S_BYTE
- U_BYTE
- S_WORD
- U_WORD
- S_LONGWORD
- U_LONGWORD
- F_FLOATING
- STRING
- ARRAY

3.3.1 Structured Data

Not all available data formats will work in a format of STRUCTURE. Table 3–13 lists the supported formats for structured data. See Section 3.3.2 for more restrictions on data formats.

Table 3–13 Data Formats Supported for Structured Data

Format	Restrictions
BIT	When writing, the use of the BIT or BOOLEAN format in structures must be in contiguous multiples of eight (8). The length of the structure containing a bit field must be an even number of target elements.
ARRAY[n]:BIT	When writing, the array size must be a multiple of 16 for PLC-5, PLC-3, PLC-2 and SLC family devices. The array size must be less than or equal to 16 for the PLC-5/250.
S_BYTE	The size must be an even multiple of 2 bytes or characters.
U_BYTE	The size must be an even multiple of 2 bytes or characters.
U_WORD	
S_WORD	
U_LONGWORD	
S_LONGWORD	
F_FLOAT	
STRING:n	The size must be an even multiple of 2 bytes or characters.
ARRAY[n]:	The size of the array must be an even multiple of the target PLC data type. For example: ARRAY[3]:S_WORD for a Longword target would not be valid. The forth word would have to be padded out by making the array size 4 or adding other members to a structure.

3.3.2 Restrictions to Data Formats

This section describes restrictions and limitations to data format usage.

- **BIT** — For all families except the PLC-5/250, the only data formats which are valid for bit writes are single bits or bit arrays with sizes that match that of the target element. For example the only valid formats for writing to a PLC-5 binary file would be:

  ```
  /FORM=BIT/NUMBER=1
  /FORM=BIT/NUMBER=16
  /FORM=ARRAY[16]:BIT
  ```

  The use of the BIT format in structures must be in contiguous multiples of eight (8). A field can contain less than 8 bits as long as there is another bit field (or bit fields) contiguous to it that makes the total 8 bits. For writes, The length of the structure must be an even number of target elements. For reads, the last field in a structure can contain less than 8 bits, or if all fields in the structure or BIT format, than the fields can contain an arbitrary number of bits.

  Bit writes greater than 1 but less than 16 bits are allowed for PI (PLC-5/250) devices.

  Bit writes cannot span element boundaries, unless the bit array is an even multiple of the element size.

- **BYTE** — Allen-Bradley PLCs are word-oriented devices. The byte formats are only valid in the context of a structure.

The byte formats may be used on any file type; the restriction is that the number of bytes used in the format must be a multiple of the PLC data element being accessed. Normally this is two for integer and binary files and Longword files are four. Other structured PLC elements such as timers and counters vary. Refer to the Allen-Bradley documentation for more information.

- **STRING** — The BASEstar Classic STRING data type may be used for any PLC file type with the following restrictions:
  - When used within a STRUCTURE, the length must be a multiple of two characters
  - When used to access string section files (ST) on PLC-5/250, PLC-5 or SLC family devices, the maximum length of the string is 82 bytes.
  - When accessing string section files, lengths smaller than 82 may be used. However an error is returned if the actual length of a string on the PLC is longer that specified in the STRING format.
  - The DAS for AB INTERCHANGE software expects the bytes to be in Allen-Bradley byte swapped order.
- **F_FLOATING** — The BASEstar Classic data type F_FLOATING may be used on PLC floating point files with no restrictions. There are, however, some behaviors with the BASEstar Classic F_FLOATING format and PLC floating point files that should be taken into consideration.

  When using the F_FLOATING format on PLC files other than floating, the value is converted to and from an integer, thereby truncating the fractional part. For example: writing the value of 1.324E+2 to an integer address using the F_FLOATING format results in the integer value of 132 being placed at the address. Reading it back, using the same format, results in 1.320E+2 being returned.

  When using the F_FLOATING format within a STRUCTURE data type, the DAS for AB INTERCHANGE software assumes the data on the PLC is formatted in the IEEE floating format. If this is not the case, data could be incorrectly converted when reading or writing.
- **STRUCTURE** — No data conversion takes place by the INTERCHANGE software when data is defined using a structure format. Fields in the structure defined as floating point are still converted, but all other fields are transferred as is. Therefore reading data as "STRUCTURE(ARRAY[2]:LONG)" in the integer file, for example, will result in a different result than reading data as "ARRAY[2]:LONG". In the first case, four words of data are transferred. In the second case, only two words of data are transferred.

3.4 Automatic Data Collection

The following section describe unsolicited data collection and pollset for the DAS for AB INTERCHANGE software.

3.4.1 Unsolicited Data Collection

BASEstar Classic device connection management can receive data values generated by plant devices without issuing a request. To use this unsolicited data collection capability, the devices and physical points must be correctly defined using BASEstar Classic software. For information on defining devices and data for unsolicited collection, refer to the *BASEstar Classic Command Line Interface User's Guide*.

Allen-Bradley devices can send unsolicited messages to the VAX host. To enable unsolicited messaging, define the device (and any required physical points) as unsolicited and enable the device, as follows:

```
DCM> MODIFY DEVICE device_name /UNSOLICITED/ENABLE
```

The DAS posts an unsolicited read request to the device. When the device sends an unsolicited message to the DAS, the read operation completes and the DAS forwards the message to BASEstar Classic device connection management software.

Messages sent from the PLC's and received by the DAS through the INTERCHANGE software are identified by an identifier. This identifier is referred to in INTERCHANGE documentation as the application address. In BASEstar Classic software, this is the unsolicited identifier, whose format is an octal value ranging from 0 to 77777.

Unsolicited data may be sent from multiple devices with the same ID, **only** if the size and format are consistent. The INTERCHANGE software will reject unsolicited messages that are inconsistent. The maximum unsolicited message size is 2000 bytes for local PLC-5/250 or 236 bytes for devices connected to a Pyramid Integrator via Data Highway.

The DAS for AB INTERCHANGE software supports only unique unsolicited identifiers defined for an individual device.

Compaq recommends that the address of the phypoint match the PLC-5 Data Table Address of the message instruction.

The Allen-Bradley PLC-5/250 and PLC-5 family sends unsolicited data to the VAX and through the INTERCHANGE software by means of the message (MSG) instruction. The values of some elements in the message instruction are critical for successfully sending data from the PLC to the DAS. The critical definitions in the message instruction are listed in Table 3–14 and Table 3–15.

Table 3–14 PLC-5 MSG Instruction for Unsolicited Data

Command field	Value
Communication Command:	PLC-2 UNPROTECTED WRITE
Size in elements:	Must match the format of the unsolicited point
Destination Data Table Address	The unsolicited ID value
Local/Remote	Local
Local Node Address	DH/DH+ address of the EI module

Table 3–15 PLC-5/250 MSG Instruction for Unsolicited Data

Command field	Value
Module ID	CLASS= UAM
Local/Remote	Local
Local Node Address	DH/DH+ address of the EI module
Communication Command:	PLC-2 UNPROTECTED WRITE
External Data Addr	The unsolicited ID value
Requested Size:	Must match the format of the unsolicited point

For more specific information on how to program the PLC's, refer to the Allen-Bradley INTERCHANGE reference and the programming manuals for the PLC-5 /250 and PLC-5.

The following is an example of an unsolicited phypoint definition. This phypoint is a 100 word array starting at address $N17:0 on device DTL5. The unsolicited ID is 200 octal.

```
DCM> DEFINE PHYPOINT -
                        /DEV=DTL5 -
                        /ADDR="$N17:0" -
                        /FORM=ARRAY[100]:WORD -
                        /UNSOL=200 -
                        /COLLECT
```

The corresponding message instruction definition for a PLC-5 would appear as follows:

```
      MESSAGE INSTRUCTION DATA ENTRY FOR CONTROL BLOCK MG10:1

              Communication Command:             PLC-2 UNPROTECTED WRITE
              PLC-5 Data Table Address:          N17:0
              Size in Elements:                  100
              Local/Remote:                      LOCAL
                 Remote Station:                 N/A
                 Link ID:                        N/A
                 Remote Link Type:               N/A
              Local Node Address:                01
              Destination Data Table Address:    200
              Port Number:                       1A
```

In this configuration, the PLC-5 is connected to the PI through the DH/DH+ port 1A.

The corresponding message instruction definition for a PLC-5/250 (with address $0N17:0) would appear as follows:

```
                          CONTROL BLOCK 0MSG0:1

  F1  Module ID:                Class   = UAM    Pw#   = 1     Port# = 1
  F2  Priority:                 HIGH
  F3  Local/Remote:             LOCAL
  F4  Local Link Type:          DH+
  F5  Remote Link Type:         N/A
  F6  Station ID:               Node#   = 0
  F7  Communication Command:    PLC-2 UNPROTECTED WRITE
  F8  Internal Data Table Addr: 0N17:0
  F9  External Data Table Addr: 0200
      Parameters:               N/A
  F10 Requested Size:           100
```

In this configuration, the PLC-5/250 is in the Pyramid Integrator chassis with the Ethernet Interface Module and the DH+ station ID is 0.

Consult the relevant Allen-Bradley documenation for information on message instructions using PLC-2, PLC3 and SLC family devices.

There is a restriction concerning the assignment of the unsolicited identifiers to DAS for AB INTERCHANGE software phypoints. The Ethernet Interface only allows one destination for an unsolicited ID. All messages coming from networked PLCs with a particular message ID are routed to only one destination on the TCP/IP network. If a INTERCHANGE application and a BASEstar Classic unsolicited point attempt to request data for the same ID, only one will be successful; the other will be returned with an error. The same is true for multiple BASEstar Classic systems. Only one BASEstar Classic system can use an unsolicited ID if they are going through the same Ethernet Interface. Figure 3–1 and Figure 3–2 show a valid and an invalid configuration.

Figure 3–1 Valid Unsolicited ID Configuration

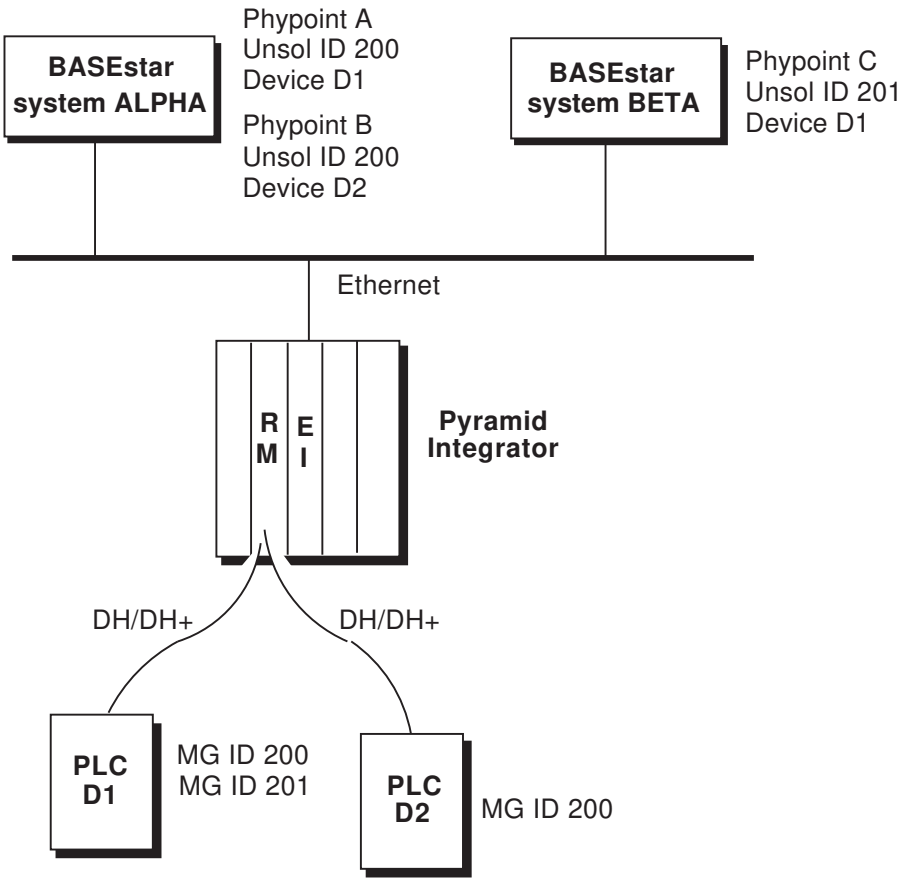


Figure 3–1 illustrates a valid unsolicited configuration. The BASEstar Classic system ALPHA has two unsolicited physical points with the same unsolicited ID of 200 octal. The configuration is valid only because the two physical points are on different devices and their format and sizes are identical. BASEstar Classic system BETA also has a physical point from device D1; because of the restriction, however, a different unsolicited ID must be used.

Figure 3–2 Invalid Unsolicited ID Configuration

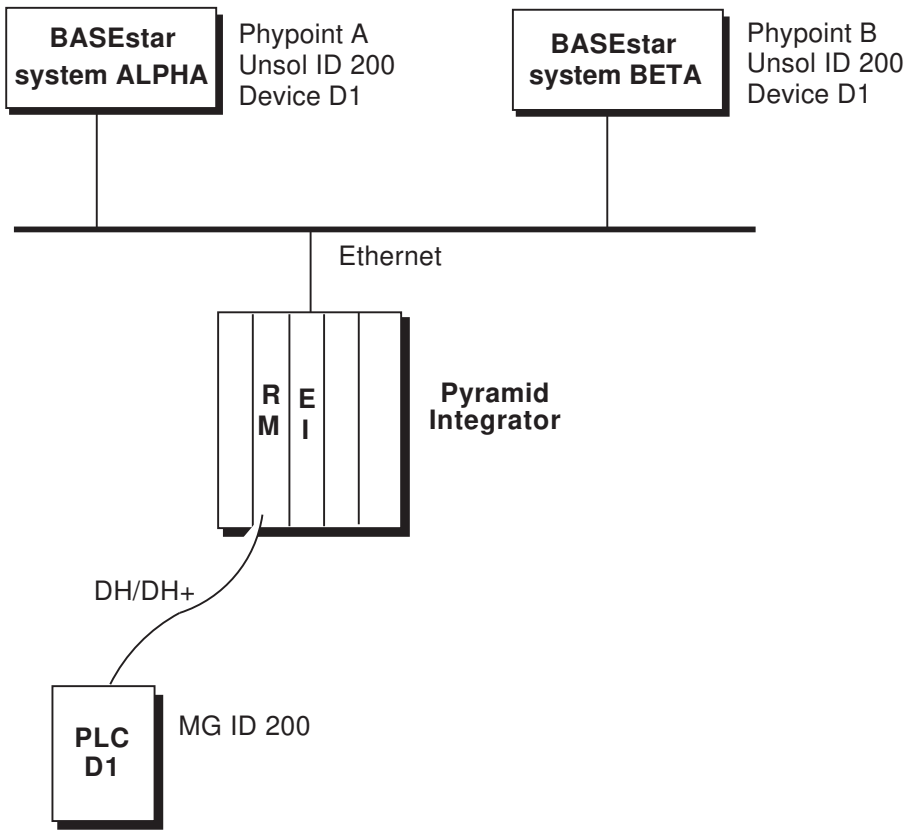


Figure 3–2 illustrates an invalid unsolicited configuration. Here both BASEstar Classic systems are attempting to use the same unsolicited ID (that both phypoints are on the same device is irrelevant). This is an invalid configuration because both BASEstar Classic systems need to route through the same Ethernet Interface module to access the unsolicited data.

3.4.2 Pollsets

BASEstar Classic device connection management physical points may be grouped together into pollsets to optimize data collection. Pollsets created for the Allen-Bradley devices have the following limitations:

- Physical points of differing data types cannot be part of the same pollset. For example, a physical point of type WORD and a physical point of type BIT cannot be members of the same pollset. Physical points of the same data type with varying element counts can reside in the same pollset.
- The maximum solicited read/write operation is 2048 bytes for local PLC-5 /250 or PLC-5/XXE devices or 242 bytes for devices connected to a Pyramid Integrator via Data Highway. The size of any pollset cannot exceed these maximum contiguous bytes limits, regardless of the number of physical points in the pollset. Therefore, the first byte of the physical point with the lowest address cannot be more than the maximum bytes away from the last byte of the physical point with the highest address.
- Points that are defined as format structure can be polled, provided that the structures of all the points in the pollset are *identical*.

- Members of pollsets created for PLC-5, SLC or PLC-3 families must have identical file types and file numbers.
- PLC-5/250 pollsets members must have identical module numbers.
- PLC-3 pollsets members must have identical contexts and must be the data table subarea.

For more information on physical points, refer to the *BASEstar Classic Command Line Interface User's Guide*.

3.5 Supporting Utilities

Three utilities are provided with BASEstar Classic DAS for Allen-Bradley INTERCHANGE software. These utilities provide logistical and maintenance support for the DAS. These utilities are:

- BCC$SYSTEM:DCM_DTL$SRV_STOP - stop server process
- BCC$SYSTEM:DCM_DTL$IMPORT_6200 - import 6200 Series Programming software files
- BCC$SYSTEM:DCM_DTL$EXPORT_6200 - export 6200 Series Programming Software files

These utilities will be discussed in the following sections.

3.5.1 Stop Server Utility

This utility is used to stop one or more of the INTERCHANGE server process if that should be come necessary.

There are some instances where anomalies outside of the control of the DAS for AB INTERCHANGE software cause a INTERCHANGE server process to no longer function properly. In most cases, this is cleared up by shutting down the server process and starting it up again.

Stopping a server is done by sending a shutdown message. It should be noted that if the server is not in a state where it can process messages, it may be necessary to kill the process in some other way ($STOP/ID=).

Example 3–10 provides an example of how to use the stop INTERCHANGE Server utility.

Example 3–10 Stop Server Utility

```
$ DTLSRV_STOP:==$BCC$SYSTEM:DCM_DTL$SRV_STOP
$ DTLSRV_STOP srv_id
```

Where:

srv_id is the number of the server or *.

The number of the server for a particular device can be obtained by looking at the BASEstar Classic history file or by doing a DCM> READ STATUS on the device. The INTERCHANGE server logs a message correlating the server with the Ethernet interface it is connecting to.

An asterisk (*) will send a shutdown message to all known servers.

3.5.2 Supporting Utilities for Allen-Bradley 6200 Series Software

Allen-Bradley 6200 Series Programming Software for the PLC-5 family provides online and off line programming of PLC-5, as well as upload, download, and compare operations. Allen-Bradley supports this software on both OpenVMS and DOS operating systems. 6200 software also supports the transfer of files between VAX systems and DOS-based T50 programming terminals using KERMIT.

Allen-Bradley 6200 programming software organizes and manipulates a PLC-5's memory image in multiple files. BASEstar Classic device connection management software performs uploads, downloads, and compares using only a single file.

Because of this incompatibility, the DAS for AB INTERCHANGE software provides two utilities to transfer compatible files between the Allen-Bradley 6200 programming software and the DAS for AB INTERCHANGE software environments. The following sections describe how to use these utilities.

3.5.2.1 Import 6200 File Utility

The Import 6200 File Utility copies a PLC-5 memory image from a 6200 Programming software environment into a DAS for AB INTERCHANGE software downloadable file.

Example 3–11 provides an example of how to use the Import 6200 File utility.

Example 3–11 Import 6200 File Utility

```
$ IMPORT6200:==$BCC$SYSTEM:DCM_DTL$IMPORT_6200
$ IMPORT6200 6200_file dcm_file
```

Where:

6200_file is the file name of the files produced in the set up.
dcm_file is the name of the BASEstar Classic downloadable file. By convention, use the file extension .LOGIC_PROGRAM

Three 6200 files are required. They must have the same file name and their extensions must be .AF5, .D1$, and .P1$. If these files are not in the same directory, or one or more is missing, the import utility will fail.

3.5.2.2 Export 6200 File Utility

The Export 6200 File Utility creates the necessary 6200 Programming Software files from a DAS for AB INTERCHANGE software downloadable file.

Example 3–12 Export 6200 File Utility

```
$ EXPORT6200:==$BCC$SYSTEM:DCM_DTL$EXPORT_6200
$ EXPORT6200 dcm_file 6200_file
```

Where:

dcm_file is the name of a file that was uploaded from BASEstar Classic software.
6200_file is the file name for the 6200 compatible files.

Three files will be created by the export utility. The utility provides the directory and file names, as well as file extensions of .AF5, .D1$, AND .P1$.

A

Allen-Bradley PLC-5/250 Addressing

Table A–1 provides the addressing mnemonics for Allen-Bradley PLC-5/250 models.

Table A–1 Supported PLC-5/250 Addresses and Data Types

Data Section	PLC Address Format	PLC Format	Byte Size
Outputs	$O:000/000 - $O377/15	Unsigned Word	2
Inputs	$I000/00 - $I377/15	Unsigned Word	2
Forced internal Storage	$IS:000 - $IS:777	Signed Word	2
Binary	$nB0000:0000/00 - $nB9999:9999/15	Signed Word	2
Integer	$nN0000:0000/00 - $nN9999:9999/15	Signed word	2
Long Integer	$nL0000:0000/00 - $nL9999:9999/15	Signed long	4
Floating point	$nF0000:0000/00 - $nF9999:9999/15	Floating Point[1]	4
Addapter Status	$AS00:0,1,2,3 - $AS37:0,1,2,3	Structure	4
System Public Status	$0S:0 - $0S:350	Word	2
Module Public Status	$nS:0 - $nS:20	Word	2
Timer structure	$nT0000:0000/00 - $nT9999:9999/15	Structure	12

Mnemonic	Instruction	Type
.EN	Enable	Bit
.TT	Timing	Bit
.DN	Done	Bit
.PRE	Preset Value	Long
.ACC	Accumulated Value	Long

[1] In IEEE 32 bit floating point format.

(continued on next page)

Table A–1 (Cont.) Supported PLC-5/250 Addresses and Data Types

Data Section	PLC Address Format			PLC Format	Byte Size
Counter Structure	$nC0000:0000/00 - $nC9999:9999/15			Structure	6
	Mnemonic	**Instruction**	**Type**		
	.CU	Up Enable	Bit		
	.CD	Down Enable	Bit		
	.DN	Done	Bit		
	.OV	Overflow	Bit		
	.UN	Underflow	Bit		
	.PRE	Preset Value	Word		
	.ACC	Accumulated Value	Word		
String	$nST0000:0000/00 - $nST9999:9999/15			Structure[2]	84
	Mnemonic	**Instruction**	**Type**		
	.LEN	Length	Word		
	.DATA[ ]	Position	82 Bytes		
Control	$nR0000:0000/00 - $nR9999:9999/15			Structure	6
	Mnemonic	**Instruction**	**Type**		
	.EN	Enable	Bit		
	.WU	Enable Unloading	Bit		
	.DN	Done	Bit		
	.EM	Empty	Bit		
	.ER	Error	Bit		
	.UL	Unload	Bit		
	.IN	Inhibit Comparisons	Bit		
	.FD	Found	Bit		
	.LEN	Length	Word		
	.POS	Position	word		

[2]Data in the .DATA portion of the structure is byte swaped.

(continued on next page)

Table A–1 (Cont.) Supported PLC-5/250 Addresses and Data Types

Data Section	PLC Address Format	PLC Format	Byte Size
Message Control	$nMSG0000:0000 - $nMSG9999:9999/15	Structure	112

Mnemonic	Instruction	Type
.EN	Enable	Bit
.ST	Start Transmision	Bit
.AD	Asynchronous Done	Bit
.AE	Asynchronous Error	Bit
.CO	Continuous	Bit
.EW	Enabled Waiting	Bit
.DN	Done	Bit
.ER	Error	Bit
.ERR	Error Code	Word
.RLEN	Requested Length	Word
.DLEN	Done Length	Word
.DATA	Reserved/Interim Use	52 Words

PID Control	$nPD0000:0000/00 - $nPD9999:9999/15	Structure	164

Mnemonic	Instruction	Type
.EN	Enable	Bit
.CT	Cascaded Type	Bit
.CL	Cascaded Loop	Bit
.PVT	PV Tracking	Bit
.DO	Derivative Of	Bit
.SWM	Software A/M Mode	Bit
.CA	Control Action	Bit
.MO	Mode	Bit
.PE	PID Equation	Bit
.INI	PID Initialized	Bit
.SPOR	SP Out of Range	Bit
.OLL	Output Limit Low	Bit
.OLH	Output Limit High	Bit
.EWD	Error Within Deadband	Bit
.DVNA	Deviation High Alarm	Bit
.DVPA	Deviation Low Alarm	Bit
.PVLA	PV Low Alarm	Bit
.PVHA	PV High Alarm	Bit

(continued on next page)

Table A–1 (Cont.) Supported PLC-5/250 Addresses and Data Types

Data Section	PLC Address Format	PLC Format	Byte Size
PID Control	$nPD0000:0000/00 - $nPD9999:9999/15	Structure	164

Mnemonic	Instruction	Type
.SP	Setpoint	Floating
.KP	Proportional Gain	Floating
.KI	Integral Gain	Floating
.KD	Derivative Time	Floating
.BIAS	Output Bias %	Floating
.MAXS	Setpoint Maximum	Floating
.MINS	Setpoint Minimum	Floating
.DB	Deadband	Floating
.SO	Set Output %	Floating
.MAXO	Output Limit High %	Floating
.MINO	Output Limit Low %	Floating
.UPD	Update Time	Floating
.PV	Process Variable	Floating
.ERR	Error	Floating
.OUT	Output	Floating
.PVH	PV Alarm High	Floating
.PVL	PV Alarm Low	Floating
.DVP	Deviation Alarm +	Floating
.DVN	Deviation Alarm -	Floating
.PVDB	PV Deadband	Floating
.MAXI	Input Range Maximum	Floating
.MINI	Input Range Minimum	Floating
.TIE	Tieback %	Floating
.ADDR	Address of Master Loop %	bytes
.DVDB	Deviation Alarm Deadband	Floating
.DATA[ ]	Reserved/Interim Use	14 Floating

B

Allen-Bradley PLC-5 Family Addressing

Table B–1 provides the addressing mnemonics for Allen-Bradley PLC-5 family models.

Table B–1 Supported PLC-5 Family Addresses and Data Types

Data Section	PLC Address Format	PLC Format	Byte Size
Outputs	$O0/0 - $O37/17 ($O277/17 for new models)	Unsigned Word	2
Inputs	$I0/0 - $I37/17 ($I1277/17 for new models)	Unsigned Word	2
ASCII[2]	$A:3/0 - $A999:999/15	Signed Word	2
BCD	$D?:0/0 - $D999:999/15	Signed Word	2
Binary	$B3:0/0 - $B999:999/15	Signed Word	2
Integer	$N7:0/0 - $N999:999/15	Signed word	2
Floating point	$F8:0/0 - $F999:999/15	Floating Point[1]	4
System Public Status	$S:0/0 - $S:128/15	Word	2
Timer structure	$T4:0 - $T999:999	Structure	6

Mnemonic	Instruction	Type
.EN	Enable	Bit
.TT	Timing	Bit
.DN	Done	Bit
.PRE	Preset Value	Word
.ACC	Accumulated Value	Word

Counter Structure	$C5:0 - $C999:999	Structure	6

Mnemonic	Instruction	Type
.CU	Up Enable	Bit
.CD	Down Enable	Bit
.DN	Done	Bit
.OV	Overflow	Bit
.UN	Underflow	Bit
.PRE	Preset Value	Word
.ACC	Accumulated Value	Word

[1] In IEEE 32 bit floating point format.

[2] Not supported on PLC-5/10, -5/12, -5/15, -5/25 models.

(continued on next page)

Table B–1 (Cont.) Supported PLC-5 Family Addresses and Data Types

Data Section	PLC Address Format	PLC Format	Byte Size
Control	$R6:0 - $R999:999	Structure	6

Mnemonic	Instruction	Type
.EN	Enable	Bit
.EU	Enable Unloading	Bit
.DN	Done	Bit
.EM	Empty	Bit
.ER	Error	Bit
.UL	Unload	Bit
.IN	Inhibit Comparisons	Bit
.FD	Found	Bit
.LEN	Length	Word
.POS	Position	word

Message[2]	$MG9:0 - $MG999:584	Structure	112

Mnemonic	Instruction	Type
.NR	No Response	Bit
.EN	Enable	Bit
.TO	Timeout	Bit
.ST	Start	Bit
.AD	Async done	Bit
.AE	Async error	Bit
.CO	Continuous	Bit
.EW	Enable waiting	Bit
.DN	Done	Bit
.ER	Error	Bit
.ERR	Error	Word
.RLEN	Receive length	Word
.DLEN	Done length	Word

[2]Not supported on PLC-5/10, -5/12, -5/15, -5/25 models.

(continued on next page)

Table B–1 (Cont.) Supported PLC-5 Family Addresses and Data Types

Data Section	PLC Address Format	PLC Format	Byte Size
PID Control[2]	$PD9:000 - $PD999:398	Structure	164

Mnemonic	Instruction	Type
.EN	Enable	Bit
.CT	Cascaded Type	Bit
.CL	Cascaded Loop	Bit
.PVT	PV Tracking	Bit
.DO	Derivative Of	Bit
.SWM	Software A/M Mode	Bit
.CA	Control Action	Bit
.MO	Mode	Bit
.PE	PID Equation	Bit
.INI	PID Initialized	Bit
.SPOR	SP Out of Range	Bit
.OLL	Output Limit Low	Bit
.OLH	Output Limit High	Bit
.EWD	Error Within Deadband	Bit
.DVNA	Deviation High Alarm	Bit
.DVPA	Deviation Low Alarm	Bit
.PVLA	PV Low Alarm	Bit
.PVHA	PV High Alarm	Bit

[2]Not supported on PLC-5/10, -5/12, -5/15, -5/25 models.

(continued on next page)

Table B–1 (Cont.) Supported PLC-5 Family Addresses and Data Types

Data Section	PLC Address Format	PLC Format	Byte Size
PID Control[2]	$PD9:000 - $PD999:398	Structure	164

Mnemonic	Instruction	Type
.SP	Setpoint	Floating
.KP	Proportional Gain	Floating
.KI	Integral Gain	Floating
.KD	Derivative Time	Floating
.BIAS	Output Bias %	Floating
.MAXS	Setpoint Maximum	Floating
.MINS	Setpoint Minimum	Floating
.DB	Deadband	Floating
.SO	Set Output %	Floating
.MAXO	Output Limit High %	Floating
.MINO	Output Limit Low %	Floating
.UPD	Update Time	Floating
.PV	Process Variable	Floating
.ERR	Error	Floating
.OUT	Output	Floating
.PVH	PV Alarm High	Floating
.PVL	PV Alarm Low	Floating
.DVP	Deviation Alarm +	Floating
.DVN	Deviation Alarm -	Floating
.PVDB	PV Deadband	Floating
.MAXI	Input Range Maximum	Floating
.MINI	Input Range Minimum	Floating
.TIE	Tieback %	Floating
.ADDR	Address of Master Loop %	bytes
.DVDB	Deviation Alarm Deadband	Floating
.DATA	Reserved/Interim Use	14 Floating

SFC status[2]	$SC9:0/0 - $SC999:999/15	Structure	6

Mnemonic	Instruction	Type
.EN	Enable	Bit
.SA	Scan active	Bit
.FS	First Scan	Bit
.LS	Last Scan	Bit
.OV	Timer overflow	Bit
.ER	Step errored	Bit
.TIM	Active Time	Word
.PRE	Preset Value	Word

[2]Not supported on PLC-5/10, -5/12, -5/15, -5/25 models.

(continued on next page)

Table B–1 (Cont.) Supported PLC-5 Family Addresses and Data Types

Data Section	PLC Address Format	PLC Format	Byte Size
String	$ST0:0 - $ST999:779	Structure[2]	84

Mnemonic	Instruction	Type
.LEN	Length	Word
.DATA	Position	82 Bytes

[2]Not supported on PLC-5/10, -5/12, -5/15, -5/25 models.

C

Allen-Bradley SLC Family Addressing

Table C–1 provides the addressing mnemonics for Allen-Bradley SLC family models.

Table C–1 Supported SLC Family Addresses and Data Types

Data Section	PLC Address Format	PLC Format	Byte Size
Outputs	$O0.0/0 - $O30.255/15	Unsigned Word	2
Inputs	$I0.0/0 - $I30.255/15	Unsigned Word	2
ASCII[1]	$A:?/0 - $A255:255/15	Signed Word	2
BCD	$D?:0/0 - $D255:255/15	Signed Word	2
Binary	$B3:0/0 - $B255:255/15	Signed Word	2
Integer	$N7:0/0 - $N255:255/15	Signed word	2
Floating point[2]	$F8:0/0 - $F255:255/15	Floating Point[1]	4
System Status	$S:0/0 - $S:?[3]/15	Word	2
Timer structure	$T4:0 - $T255:255	Structure	6

Mnemonic	Instruction	Type
.EN	Enable	Bit
.TT	Timing	Bit
.DN	Done	Bit
.PRE	Preset Value	Word
.ACC	Accumulated Value	Word

[1]Not available on SLC 500, SLC 5/01, SLC 5/02 and SLC 5/03 processors.

[2]In IEEE 32 bit floating point format.

[3]Processor dependent.

(continued on next page)

Table C–1 (Cont.) Supported SLC Family Addresses and Data Types

Data Section	PLC Address Format			PLC Format	Byte Size
Counter Structure	$C5:0 - $C255:255			Structure	6
	Mnemonic	**Instruction**	**Type**		
	.CU	Up Enable	Bit		
	.CD	Down Enable	Bit		
	.DN	Done	Bit		
	.OV	Overflow	Bit		
	.UN	Underflow	Bit		
	.UA[4]	Update Accumulator	Bit		
	.PRE	Preset Value	Word		
	.ACC	Accumulated Value	Word		
Control	$R6:0 - $R255:255			Structure	6
	Mnemonic	**Instruction**	**Type**		
	.EN	Enable	Bit		
	.EU	Enable Unloading	Bit		
	.DN	Done	Bit		
	.EM	Empty	Bit		
	.ER	Error	Bit		
	.UL	Unload	Bit		
	.IN	Inhibit Comparisons	Bit		
	.FD	Found	Bit		
	.LEN	Length	Word		
	.POS	Position	word		
String[1]	$ST0:0 - $ST999:779			Structure	84
	Mnemonic	**Instruction**	**Type**		
	.LEN	Length	Word		
	.DATA	Position	82 Bytes		

[1]Not available on SLC 500, SLC 5/01, SLC 5/02 and SLC 5/03 processors.

[4]Available only in SLC 500 fixed style processors equipped with a High Speed Counter.

D
Logged Messages

The messages in the following sections are logged to the BASEstar Classic history file by the DAS for AB INTERCHANGE software. These messages are logged to provide more detailed diagnostic information than what is supplied by the returned status values.

Messages logged to the history file for DAS for AB INTERCHANGE software use event class 21 and event type 2.

To view all messages logged by this DAS use the following syntax:

```
$ BSTAR
BSTAR> SHOW HISTORY/EVENT=21.2.*
```

D.1 NI Logged Messages

The following messages are logged by the Network Interface (NI).

Connection made with server DCM_DTL$SRV_!UL.

Explanation: Success. A connection has been made by the DAS to the indicated DTL server process. At this point, a connection to the actual PLC has not yet been established.

User Action: None.

Disconnect from server DCM_DTL$SRV_!UL.

Explanation: Error. The indicated server process has disconnected from the DAS. The disconnect usually is caused by an abnormal termination of the server process, but could also be because of a BASEstar shutdown or because someone intentionally shut down the server process.

User Action: Examine the log file of the server process and the history file to determine the cause of the server shutdown and then take corrective action based on the error found.

License not found for DAS for INTERCHANGE software.

Explanation: Error. A valid license hasn't been loaded for the DAS.

User Action: Obtain a valid license and load it onto the system where the DAS is running.

Starting DTL server process DCM_DTL$SRV_!UL.

Explanation: Informational. The DAS is attempting to start the DTL server process.

User Action: None.

DCM_DTL$SRV_!UL did not start in time, aborting associated connections.

Explanation: Error. The indicated server process did not start within the allotted time (59 seconds), or if the server started, it did not successfully establish a connection with the DAS.

User Action: Examine the DTL server log file and the history file to determine if any errors were logged by the server process. Take action based on the errors found.

D.2 PE Logged Messages

The following messages are logged by the Protocol Emulator (PE).

Device address string too long: !/ "!AD"

Explanation: Error. A point address is too long to fit into the space allocated for it in the DAS. The DAS allows 18 characters for the point address.

User Action: Use a shorter address for the point.

Invalid qualifier: !AD

Explanation: Error. An invalid qualifier has been specified in an upload or download request. The qualifier in error is logged.

User Action: Consult the DAS guide to determine valid qualifiers for the command being requested.

Device MODEL definition is !AD family; Actual PLC TYPES is !AD.

Explanation: Error. When reading the status from a PLC, the DAS determined that the model for the actual PLC does not match the model of the PLC definition.

User Action: Modify the device definition to match the actual model of the PLC.

Error in the device network address !AD and or model !AD.

Explanation: Error. The DAS detected an error in the model or network address for the device.

User Action: If the model is invalid, use one of the predefined DTL types when creating the device. If the network address is invalid, see the DAS guide for information on the proper syntax for a network address for this DAS.

Model !AD not supported.

Explanation: Error. The model in the device definition is not supported by the DAS.

User Action: Use one of the predefined DTL types from the DCM_DTL_ CONFIG.TEMPLATE file when creating the device.

Error converting read data for address !AD.

Explanation: Error. An error occurred when doing data conversion for the address indicated. The conversion error is probably due to incompatible data types between the request and the data in the PLC.

User Action: Use a different data type when requesting the data.

Error converting write data for address !AD.

Explanation: Error. An error occurred when doing data conversion for the address indicated. The conversion error is probably due to incompatible data types between the request and the data in the PLC.

User Action: Use a different data type when writing the data.

D.3 DTL Server Logged Messages

The following messages are logged by the DTL server process. All messages first log the server that generated the error before the actual error message is logged. INTERCHANGE software can return four types of error codes as follows:

- DTL errors. These errors are prefixed by DTL: and are returned by the INTERCHANGE software.
- "C" errors. These errors are prefixed by TCP/IP: and are returned by the "C" socket routines that implement the protocol used to communicate to the Ethernet Interface.
- Operating system errors. These errors do not have a prefix and are returned by OpenVMS. These errors generally are a result of privilege or quota problems with the DTL server process.
- Data Highway errors. These errors are prefixed by PCCC: and are returned by the Data Highway Plus protocol. These errors just log (in hex) the error code returned by Data Highway. Consult *INTERCHANGE Software Reference Manual* Appendix B for a description of the errors that can be returned.

DTL DAS server process shut down abnormally.

Explanation: Error. The server process is shutting down due to an unrecoverable error. The history file should also have logged the error that caused the shutdown.

User Action: Examine the history file for the error that caused the termination and take action based on the error found.

DTL DAS disconnected circuit from server process.

Explanation: Error. The DAS has disconnected from the server process. The cause is probably because of an abnormal shutdown of the ILAN$DEVSRV process.

User Action: Examine the history file for any errors logged that might account for the abnormal shutdown. Also examine the ILAN$DEVSRV log, error and ACCVIO files for causes for the shutdown. Restart ILAN$DEVSRV.

Error calling DTL_DIAG_STATUS.

Explanation: Error. An error occurred when requesting status information from the PLC.

User Action: Examine the error message returned and take action based on the error found.

DTL_C_DEFINE error.!/!AD

Explanation: Error. An error occurred when defining a point for reading or writing. A point must be defined before it can be read or written. The message logs the address part of the string passed to the DTL_C_DEFINE routine.

User Action: Examine the error returned by INTERCHANGE software and take action based on the error.

Error calling DTL_DOWNLOAD.

Explanation: Error. An error occurred when downloading a file to the PLC.

User Action: Examine the error returned by INTERCHANGE software and take action based on the error.

Error calling DTL_CONNECT for EI !AD.

Explanation: Error. An error occurred when attempting to connect to the indicated Ethernet Interface. A connection must be established with the Ethernet Interface before any subsequent requests can be processed.

User Action: Examine the error returned by INTERCHANGE software and take action based on the error.

Error calling DTL_DISCONNECT.

Explanation: Error. An error occurred when attempting to disconnect from the indicated Ethernet Interface.

User Action: Examine the error returned by INTERCHANGE software and take action based on the error.

Disconnected from EI !AD.

Explanation: Informational. The DTL server process successfully disconnected from the indicated Ethernet Interface.

User Action: None.

DTL_READ error. Address: !AD.

Explanation: Error. An error occurred when attempting to read data from a PLC. The data table address from which the read was attempted is also logged.

User Action: Examine the error returned by INTERCHANGE software and take action based on the error.

DTL_UNDEF error. Address: !AD.

Explanation: Error. An error occurred when attempting to undefine an address. The address that was being undefined is logged in the message. Solicited points are undefined when the number of solicited points (controlled by ILAN$DTL_MAX_ELEMENTS) is exceeded.

User Action: Examine the error returned by INTERCHANGE software and take action based on the error.

DTL_WRITE error. Address: !AD.

Explanation: Error. An error occurred when attempting to write data to a PLC. The data table address to which the write was attempted is also logged.

User Action: Examine the error returned by INTERCHANGE software and take action based on the error.

Established connection to EI !AD.

Explanation: Informational. The DTL server process successfully connected to the indicated Ethernet Interface. A connection must be established to the Ethernet Interface before any subsequent operations can take place.

User Action: None.

Lost connection to EI !AD.

Explanation: Error. The connection to the indicated Ethernet Interface has been broken. The most likely causes are that the PLC has been powered down or there has been a break in the network connection to the PLC.

User Action: Examine the error returned by INTERCHANGE software and take action based on the error.

Unsolicited exception. Unsol ID: !OW.

Explanation: Error. An error occurred when receiving unsolicited data. The most likely cause of this error is that two hosts have defined points with the same unsolicited ID on the same PLC.

User Action: Examine the error returned by INTERCHANGE software and take action based on the error.

DTL DAS server process running.

Explanation: Informational. The server process was started successfully and is now running.

User Action: None.

Max queued for this run: !UL.

Explanation: Informational. This message is logged when the server process is shut down. It logs the maximum number of requests that were queued (i.e. requested but not yet processed) since the DTL server process was first started. If this number is large, then more requests are being received than can be processed by the PLC.

User Action: None.

DTL server process shut down.

Explanation: Informational. The DTL server process has successfully finished shutting down.

User Action: None.

NOS error X!XL.

Explanation: Error. A Data Highway Plus error occurred that was not translated by the INTERCHANGE software into a DTL error code.

User Action: See Appendix B of the *INTERCHANGE Software Reference Manual* for a description of the error code and take action based on the error.

Unsolicited write. No unsolicited definition found for ID: !OW.

Explanation: Error. Unsolicited data was received, but the DTL server process has no record of requesting unsolicited data for the indicated unsolicited ID. This error is most likely to occur when a device is disabled and the unsolicited point has been cancelled, but the Ethernet Interface is still sending unsolicited data to the host.

User Action: If the problem persists, the PLC may need to be reset to cancel the sending of the unsolicited data. If the problem does not persist, then no action needs to be taken.

DTL server process shutting down.

Explanation: Informational. The DTL server process has received a request to shut down and is starting the shutdown process.

User Action: None.

DTL_UNSOL_DEF error defining ID: !OW on EI: !AD.

Explanation: Error. An error occurred when attempting to define an unsolicited point. Unsolicited points must be defined before unsolicited data will be sent by the Ethernet Interface.

User Action: Examine the error returned by INTERCHANGE software and take action based on the error.

DTL_UNDEF error. Unsol ID: !OW.

Explanation: Error. An error occurred when attempting to undefine an unsolicited point. Unsolicited points are undefined when a device is disabled or a point is deleted.

User Action: Examine the error returned by INTERCHANGE software and take action based on the error.

Error calling DTL_UPLOAD.

Explanation: Error. An error occurred when uploading a file from the PLC.

User Action: Examine the error returned by INTERCHANGE software and take action based on the error.

Index

A

Accessing DAS functions, 3–1
Addressing
  PLC-2 family, 3–5
  PLC-3 family, 3–4
  PLC-5/250, 3–2
  PLC-5 family, 3–2
  SLC family, 3–3

C

Configuration
  device records, 2–17
  editing configuration file, 2–11
  network addresses, 2–17
  path records, 2–15
  TCP/IP parameters, 2–11
  type records, 2–11
Configuring parameters
  BCC$MAX_PORT_QUOTA, 2–19
  ILAN$DTLSRV_PRIORITY, 2–20
  ILAN$DTL_BACKLOG, 2–20
  ILAN$DTL_CYCLE_TIME, 2–21
  ILAN$DTL_DUP_EI_ID, 2–23
  ILAN$DTL_DWNL_FLAGS, 2–21
  ILAN$DTL_MAX_ELEMENTS, 2–21
  ILAN$DTL_MAX_UNSOL_ELEMENTS, 2–21
  ILAN$DTL_SCMP_FLAGS, 2–22
  ILAN$DTL_TIMEOUT, 2–22
  ILAN$DTL_UDC_TIMEOUT, 2–22
  ILAN$DTL_UPL_FLAGS, 2–22
  ILAN$MAX_SPT_REQUESTS, 2–19

D

DAS
  See Device access software
Data collection, automatic, 3–15
  pollsets, 3–19
  unsolicited, 3–16
Data format
  restrictions, 3–14
Data type formats, 3–13
Device access software
  accessing functions, 3–1
  description of, 1–1
Device access software (cont'd)
  functions, 1–5
  installation of, 2–1
  installation requirements, 2–1
  overview, 1–1
  supported data types, 3–13
  supported devices, 1–5
  supported functions, 1–5, 3–2
  using, 3–1
Device communications, 1–1
  tracing, 2–24
Device records, 2–17
Download function, 3–8
DTL Server Process
  quotas for, 2–23

F

Files, created during installation, 2–7
Functions, 3–1
  download, 3–8
  read data, 3–5
  read status, 3–6
  upload, 3–8
  write data, 3–5

I

Installation, 2–1
  files created, 2–7
  messages, 2–9
  procedure, 2–4
Installation requirements, 2–1
  disk space, 2–3
  hardware, 2–1
  software, 2–2

M

Message port quota, 2–19
Messages
  error and return values, 1–5
  installation, 2–9
  logged, D–1

P

Path records, 2–15
Plant-floor equipment, setting up, 2–23
Pollsets, 3–19
Postinstallation tasks, 2–10
Problem reporting, 2–24
Product failure, 2–24

Q

Quotas, 2–23

R

Read data function, 3–2
Read status function, 3–6

S

SPT block usage, 2–19
Structured data, 3–13
Supported data types, 3–13

T

Trace
  using, 2–24
Type records, 2–11

U

Unsolicited data collection, 3–16
Upload function, 3–8

W

Write data function, 3–2