BASEstar™ Classic DAS for Allen-Bradley Data Highway™ Protocol

Installation and User's Guide

Order Number: AA-R210B-TE

April 2000

This manual describes how to install and use the DAS for Allen-Bradley Data Highway Protocol 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
Software Version:	BASEstar Classic DAS for Allen-Bradley Data Highway Protocol, Version 3.4A

Compaq Computer Corporation
Houston, Texas

April 2000

© 1989 Compaq Computer Corporation.

COMPAQ, VAX, and VMS 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.

SLC, SLC-500, SLC-5/01, SLC-5/02, SLC-5/03 and SLC-5/04 are trademarks of Allen-Bradley Company, Inc.

Data Highway 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

This document describes how to install and use BASEstar Classic DAS for Allen-Bradley Data Highway Protocol.

Intended Audience

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

- BASEstar Classic for OpenVMS software
- BASEstar Classic DAS for Allen-Bradley Data Highway Protocol

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 a solid understanding of OpenVMS operations and administration, as well as OpenVMS application software. In addition, knowledge of the Allen-Bradley PLCs and the specific requirements of the installation site is essential.

Document Structure

This document is organized as follows:

Document Structure

This document is organized as follows:

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

Associated Documents

Further information on topics covered in this document can be found in the following documents:

- *BASEstar Classic Installation Guide*
- *BASEstar Classic Configuration and Tuning Guide*
- *BASEstar Classic Menu Interface User's Guide*
- *BASEstar Classic Command Line Interface User's Guide*
- *BASEstar Classic Introduction to Callable Services*
- *BASEstar Classic Guide to Writing Device Access Software*
- *BASEstar Classic Application Programming Interface Reference Guide*

Information on Allen-Bradley addressing and programming can be found in the following documents.

- *Allen-Bradley Data Highway/Data Highway Plus Protocol and Command Set*
- *Pyramid Integrator System Addressing Reference*
- *PLC-5 Family Addressing Reference*
- *PLC-3 Family Addressing Reference*
- *PLC-2 Family Addressing Reference*
- *SLC 500 Family of Programmable Controllers Addressing Reference Manual*
- *PLC-5/250 Programming Software*
- *PLC-5 Programming Software*

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 Data Highway Protocol. It also briefly describes Allen-Bradley PLCs and the supported functions for the DAS for AB Data Highway Protocol.

1.1 Description

The BASEstar Classic DAS for Allen-Bradley Data Highway Protocol allows you to access Allen-Bradley PLCs through BASEstar Classic device connection management. Device connection management is the device connection/control component of BASEstar Classic for OpenVMS. BASEstar Classic software is designed to facilitate the integration of manufacturing equipment and applications, accelerate the development of integrated manufacturing systems, and provide an architecture for consistent development of manufacturing applications.

Using the DAS for AB Data Highway Protocol, users or applications can perform a variety of device access functions, including: reading and writing data, uploading and downloading logic programs, and reading the status of a device.

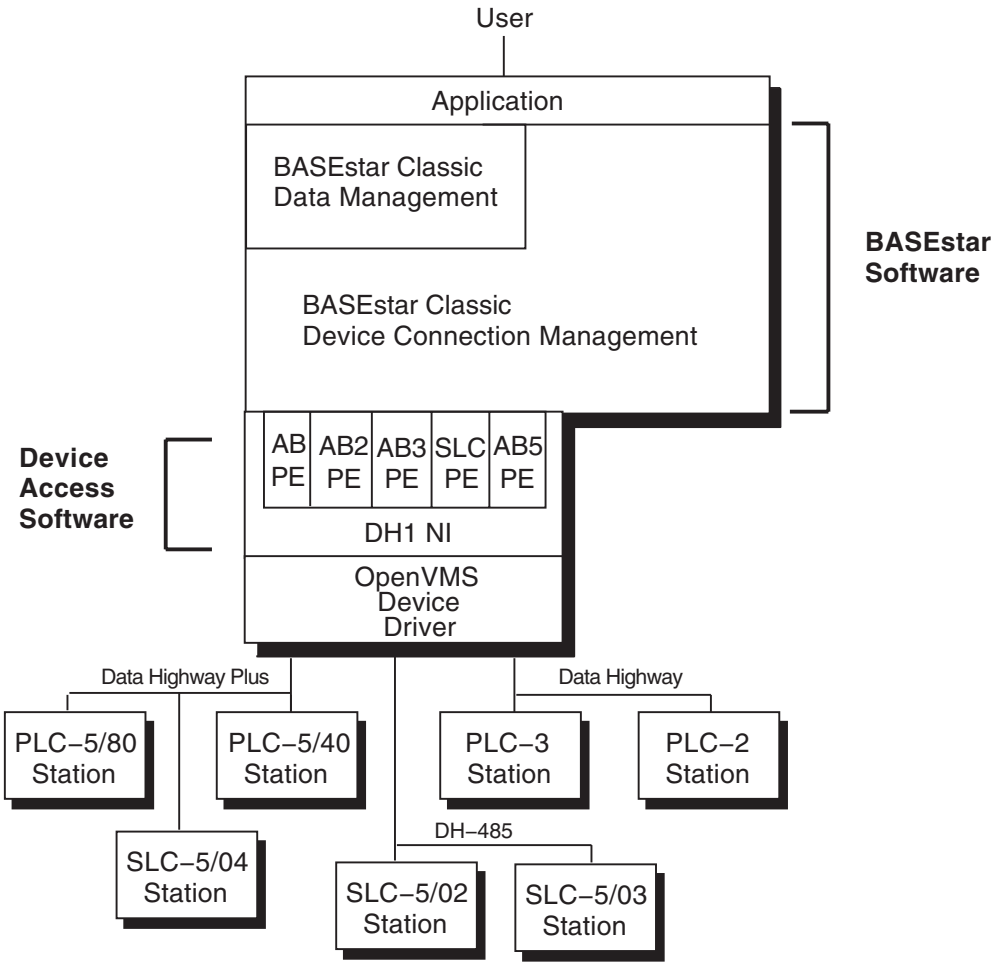
1.2 Device Communications

The DAS for AB Data Highway Protocol consists of five protocol emulators (PEs) and one network interface (NI). The PE and NI work together to provide device-specific communications for BASEstar Classic device connection management's generic callable services.

The PE translates BASEstar Classic device connection management's generic services into a device-understandable format. The PE also converts device-specific protocol into a BASEstar Classic device connection management format. The NI works directly with a OpenVMS driver to send data to and receive data from plant-floor devices.

Figure 1–1 shows how the DAS for AB Data Highway Protocol facilitates communications between device connection management and the devices.

Figure 1–1 DAS Communications



1.3 Supported Functions and Devices

You can perform only the BASEstar Classic device connection management functions that are supported by a device's PE. These functions can be accessed through BASEstar Classic library management and BASEstar Classic device connection management's menu system, commands, and callable services.

The DAS for AB Data Highway Protocol supports the following BASEstar Classic device connection management functions:

- Start and stop operations on a device
- Upload the contents of a device's memory to an OpenVMS file
- Download an OpenVMS file to a device's memory
- Read data from a specific address in a device's memory
- Write data to a specific address in a device's memory

- Read status for a device

The DAS for AB Data Highway Protocol supports a variety of devices, as shown in Table 1–1.

Table 1–1 Allen-Bradley Devices and Supported Functions

Device	Upload	Download	Start	Stop	Read Data	Write Data	Read Status
PLC-1774	X	X	X	X	X	X	X
PLC-2	X	X			X	X	X
PLC-2/02	X	X			X	X	X
PLC-2/05	X	X			X	X	X
PLC-2/15	X	X			X	X	X
PLC-2/16	X	X			X	X	X
PLC-2/17	X	X			X	X	X
PLC-2/20	X	X			X	X	X
PLC-2/30	X	X			X	X	X
PLC-3 family	X	X	X	X	X	X	X
PLC-5/10	X	X	X	X	X	X	X
PLC-5/11	X	X	X	X	X	X	X
PLC-5/12	X	X	X	X	X	X	X
PLC-5/15	X	X	X	X	X	X	X
PLC-5/16	X	X	X	X	X	X	X
PLC-5/20	X	X	X	X	X	X	X
PLC-5/25	X	X	X	X	X	X	X
PLC-5/26	X	X	X	X	X	X	X
PLC-5/30	X	X	X	X	X	X	X
PLC-5/36	X	X	X	X	X	X	X
PLC-5/40	X	X	X	X	X	X	X
PLC-5/46	X	X	X	X	X	X	X
PLC-5/60	X	X	X	X	X	X	X
PLC-5/66	X	X	X	X	X	X	X
PLC-5/80	X	X	X	X	X	X	X
PLC-5/86	X	X	X	X	X	X	X
PLC-5/250			X	X	X	X	X
SLC-500					X	X	X
SLC-5/01					X	X	X
SLC-5/02					X	X	X
SLC-5/03					X	X	X
SLC-5/04					X	X	X

For more information about the supported functions, refer to Chapter 3 of this document.

2
Installing the DAS

This chapter provides the information you need to install the DAS for AB Data Highway Protocol and to configure your system.

2.1 Installation Requirements

Review the following hardware and software requirements to ensure that your system is prepared for the DAS for AB Data Highway Protocol 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 Requirements

The following minimum hardware is required for the installation and operation of DAS for AB Data Highway Protocol components.

- Any OpenVMS system configuration that has an RS-232 serial port or LAT port connection and is valid for BASEstar Classic Clasic for OpenVMS.
- Any of the following Allen-Bradley PLCs:
  - Allen-Bradley PLC-1774
  - Allen-Bradley PLC-2, PLC-2/02, PLC-2/05, PLC-2/15, PLC-2/16, PLC-2/17, PLC-2/20, PLC-2/30
  - Allen-Bradley PLC-3 family of devices
  - Allen-Bradley PLC-5/10, PLC-5/11, PLC-5/12, PLC-5/15, PLC-5/16, PLC-5/20, PLC-5/25, PLC-5/26, PLC-5/30, PLC-5/36, PLC-5/40, PLC-5/46, PLC-5/60, PLC-5/66, PLC-5/80, PLC-5/86, PLC-5/250
  - Allen-Bradley SLC-500, SLC-5/01, SLC-5/02, SLC-5/03, SLC-5/04
- ANSI compatible video display terminal

In addition to the above named devices, an appropriate communication card must also be provided for communicating to the appropriate network:

- For DH+ networks: 1770-KF2, 1785-KE
- For DH networks: 1771-KE, -KF
- For DH-485 networks: 1770-KF3

NOTE: The SLC-500, SLC-5/01, SLC-5/02 and SLC-5/03 can be connected only to a DH-485 network. The SLC-5/04 can be connected to a Data Highway Plus or DH-485 network.

NOTE: The embedded responses feature is not supported by the DH1 NI. Embedded responses is a switch selectable option in the Data Highway Communications Adapter (1771-KE, 1771-KF, 1771-KF2) and must be disabled.

2.1.2 Software Requirements

The following Digital software must be installed prior to installing the DAS for AB Data Highway Protocol:

- 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, Release 3.4

For information on installing the above software, 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 Disk Space Requirements

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

Table 2–1 Disk Space Requirements

Peak/Net Usage	Approximate Space Requirements (Blocks)
Peak usage (during installation)	850 (VAX)
	1200 (Alpha)
Net usage (after installation)	550 (VAX)
	800 (Alpha)

2.2 Incompatibility With DAS for Allen-Bradley SLC

This DAS is incompatible with the DAS for Allen-Bradley SLC Programmable Controllers. If you have previously installed the DAS for AB SLC PLCs, devices defined using this DAS will no longer function. Specifically, the SLCDH1 NI will no longer function and the paths using the SLCDH1 NI must be replaced with paths using the DH1 NI.

2.3 Installation

When your system meets all hardware and software requirements, you can install the DAS for AB Data Highway Protocol. The installation takes from 1 to 5 minutes, depending on system load and configuration.

Install the DAS for AB Data Highway Protocol 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_ABVVA034 ddcu:
   ```

   The **DCM_ABVVA034** argument is the kit name. The **034** portion of the name is the version number. The **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:

```
        OpenVMS VAX Software Product Installation Procedure V7.2

It is 25-JAN-2000 at 15:28.

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_ABVVA V3.4

        Beginning installation of DCM_ABVVA V3.4

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

   Copyright 1989 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 Data Highway Protocol
   installation procedure.


        Product:      DAS-AB-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...

   Now checking disk space...

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

* Does your site have AB PLC-1774 PLCs [YES]?
* Does your site have AB PLC-2 family PLCs [YES]?
* Does your site have AB PLC-3 family PLCs [YES]?
* Does your site have AB PLC-5 family PLCs [YES]?
* Does your site have AB SLC family PLCs [YES]?

   The installation procedure has no further questions to ask and will
   complete in 1 to 5 minutes depending on the system and system load.

   The configuration template file for AB support, DCM_AB_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.

%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...

   Copyright 1989 Compaq Computer Corporation

   Confidential commercial computer software.  Valid license required.

   Executing the Installation Verification Procedure.

   BASEstar Classic DAS for Allen-Bradley Data Highway Protocol
   installation procedure has succeeded.

        Installation of DCM_ABVVA V3.4 completed at 15:40

        VMSINSTAL procedure done at 15:40
```

2.3.1 Files Created During Installation

Table 2–2 lists the files that the DAS for AB Data Highway Protocol installation procedure creates and the directories in which they are placed.

Table 2–2 Files Created During Installation

Directory	Filename
BCC$SYSDATA:	DCM_AB_CONFIG.TEMPLATE
	DCM_AB$MSG.DAT
SYS$LIBRARY:	ILAN_AB.EXE
	ILAN_AB2.EXE
	ILAN_AB3.EXE
	ILAN_AB5.EXE
	ILAN_SLC.EXE
	ILAN_DH1.EXE
	ILAN_AB_SCMP.EXE (VAX)
SYS$HELP	DCM_ABVVA034.RELEASE_NOTES (VAX)
	DCM_ABVAA034.RELEASE_NOTES (Alpha)

2.3.2 Installation Messages

You may see VMSINSTAL messages during the installation procedure. The following messages are specific to the DAS for AB Data Highway Protocol installation:

BADBCC, BASEstar Classic software must be installed before DAS for AB Data Highway Protocol.

Explanation: Error. Incorrect version of or missing BASEstar Classic software.

User Action: Install BASEstar Classic for OpenVMS, Release 3.4 or higher software.

BADDCM, BASEstar Classic Device Connect must be installed before the DAS for AB Data Highway Protocol.

Explanation: Error. Incorrect version of or missing BASEstar Classic DCM software.

User Action: Install BASEstar Classic DCM for OpenVMS, Release 3.4 or higher software.

BADVMS (VAX), The DAS for AB Data Highway Protocol must be installed under OpenVMS V5.5-2 or greater.

Explanation: Error. Incorrect version of OpenVMS.

User Action: Install OpenVMS V5.5-2 or higher.

BADVMS (Alpha), The DAS for AB Data Highway Protocol must be installed under OpenVMS V6.1 or greater.

Explanation: Error. Incorrect version of OpenVMS.

User Action: Install OpenVMS V6.1 or higher.

NETBLOCKS (VAX), DAS for AB Data Highway Protocol requires 550 blocks after installation.

Explanation: Error. Not enough disk space to complete installation.

User Action: Delete any unnecessary files, then reinstall.

NETBLOCKS (Alpha), DAS for AB Data Highway Protocol requires 800 blocks after installation.

Explanation: Error. Not enough disk space to complete installation.

User Action: Delete any unnecessary files, then reinstall.

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.

2.4 Postinstallation Tasks

This section describes the tasks to perform after installing the DAS for AB Data Highway Protocol including editing the configuration file, setting the device connection management support block parameter, configuring ports, and setting up plant-floor equipment.

2.4.1 Editing the Configuration File

A configuration file, BCC$SYSDATA:DCM_AB_CONFIG.TEMPLATE, is supplied with the DAS kit. The configuration file contains definitions for types, paths, and devices. A type record represents a protocol emulator (PE). A path record represents a network interface (NI). You must edit this configuration file to include site-specific information about types and paths before you execute this file.

The following sections contain examples of type, path, and device records. Refer to the *BASEstar Classic Command Line Interface User's Guide* for more information about creating type, path, and device definitions.

2.4.1.1 Editing Type Records

The following example shows the type records created by the configuration file:

```
create type AB_PLC_2_02/manufacturer=AB/model="PLC-2/02"/protocol=AB2 -
/description="Allen Bradley PLC-2/02 Programmable Controller"/log

create type AB_PLC_2_05/manufacturer=AB/model="PLC-2/05"/protocol=AB2 -
/description="Allen Bradley PLC-2/05 Programmable Controller"/log

create type AB_PLC_2_15/manufacturer=AB/model="PLC-2/15"/protocol=AB2 -
/description="Allen Bradley PLC-2/15 Programmable Controller"/log

create type AB_PLC_2_16/manufacturer=AB/model="PLC-2/16"/protocol=AB2 -
/description="Allen Bradley PLC-2/16 Programmable Controller"/log
```

```
create type AB_PLC_2_17/manufacturer=AB/model="PLC-2/17"/protocol=AB2 -
/description="Allen Bradley PLC-2/17 Programmable Controller"/log

create type AB_PLC_2_20/manufacturer=AB/model="PLC-2/20"/protocol=AB2 -
/description="Allen Bradley PLC-2/20 Programmable Controller"/log

create type AB_PLC_2_30/manufacturer=AB/model="PLC-2/30"/protocol=AB2 -
/description="Allen Bradley PLC-2/30 Programmable Controller"/log

create type AB_PLC_2_IM/manufacturer=AB/model="PLC-IM"/protocol=AB2 -
/description="Allen Bradley Interface Module"/log

create type AB_PLC_3/manufacturer=AB/model="PLC-3"/protocol=AB3 -
/description="Allen Bradley PLC-3 Programmable Controller"/log

create type AB_PLC_3_IM/manufacturer=AB/model="PLC-IM"/protocol=AB3 -
/description="Allen Bradley Interface Module"/log

create type AB_PLC_5_10/manufacturer=AB/model="PLC-5/10"/protocol=AB5 -
/description="Allen Bradley PLC-5/10 Programmable Controller"/log

create type AB_PLC_5_11/manufacturer=AB/model="PLC-5/11"/protocol=AB5 -
/description="Allen Bradley PLC-5/11 Programmable Controller"/log

create type AB_PLC_5_12/manufacturer=AB/model="PLC-5/12"/protocol=AB5 -
/description="Allen Bradley PLC-5/12 Programmable Controller"/log

create type AB_PLC_5_15/manufacturer=AB/model="PLC-5/15"/protocol=AB5 -
/description="Allen Bradley PLC-5/15 Programmable Controller"/log

create type AB_PLC_5_16/manufacturer=AB/model="PLC-5/16"/protocol=AB5 -
/description="Allen Bradley PLC-5/16 Programmable Controller"/log

create type AB_PLC_5_20/manufacturer=AB/model="PLC-5/20"/protocol=AB5 -
/description="Allen Bradley PLC-5/20 Programmable Controller"/log

create type AB_PLC_5_25/manufacturer=AB/model="PLC-5/25"/protocol=AB5 -
/description="Allen Bradley PLC-5/25 Programmable Controller"/log

create type AB_PLC_5_26/manufacturer=AB/model="PLC-5/26"/protocol=AB5 -
/description="Allen Bradley PLC-5/26 Programmable Controller"/log

create type AB_PLC_5_30/manufacturer=AB/model="PLC-5/30"/protocol=AB5 -
/description="Allen Bradley PLC-5/30 Programmable Controller"/log

create type AB_PLC_5_36/manufacturer=AB/model="PLC-5/36"/protocol=AB5 -
/description="Allen Bradley PLC-5/36 Programmable Controller"/log

create type AB_PLC_5_40/manufacturer=AB/model="PLC-5/40"/protocol=AB5 -
/description="Allen Bradley PLC-5/40 Programmable Controller"/log

create type AB_PLC_5_46/manufacturer=AB/model="PLC-5/46"/protocol=AB5 -
/description="Allen Bradley PLC-5/46 Programmable Controller"/log

create type AB_PLC_5_60/manufacturer=AB/model="PLC-5/60"/protocol=AB5 -
/description="Allen Bradley PLC-5/60 Programmable Controller"/log

create type AB_PLC_5_66/manufacturer=AB/model="PLC-5/66"/protocol=AB5 -
/description="Allen Bradley PLC-5/66 Programmable Controller"/log

create type AB_PLC_5_80/manufacturer=AB/model="PLC-5/80"/protocol=AB5 -
/description="Allen Bradley PLC-5/80 Programmable Controller"/log

create type AB_PLC_5_86/manufacturer=AB/model="PLC-5/86"/protocol=AB5 -
/description="Allen Bradley PLC-5/86 Programmable Controller"/log

create type AB_PLC_5_250/manufacturer=AB/model="PLC-5/250"/protocol=AB5 -
/description="Allen Bradley PLC-5/250 Programmable Controller"/log

create type AB_PLC_5_IM/manufacturer=AB/model="PLC-IM"/protocol=AB5 -
/description="Allen Bradley Interface Module"/log

create type AB_SLC_500/manufacturer=AB/model="SLC-500"/protocol=SLC -
/description="Allen Bradley SLC-500 Programmable Controller"/log
```

```
create type AB_SLC_5_01/manufacturer=AB/model="SLC-5/01"/protocol=SLC -
/description="Allen Bradley SLC-5/01 Programmable Controller"/log

create type AB_SLC_5_02/manufacturer=AB/model="SLC-5/02"/protocol=SLC -
/description="Allen Bradley SLC-5/02 Programmable Controller"/log

create type AB_SLC_5_03/manufacturer=AB/model="SLC-5/03"/protocol=SLC -
/description="Allen Bradley SLC-5/03 Programmable Controller"/log

create type AB_SLC_5_04/manufacturer=AB/model="SLC-5/04"/protocol=SLC -
/description="Allen Bradley SLC-5/04 Programmable Controller"/log

create type AB_SLC_IM/manufacturer=AB/model="PLC-IM"/protocol=SLC -
/description="Allen Bradley Interface Module"/log
```

The types listed above are the only types supported by the DAS. If you have a device that is not in the above list, then it is likely that you can do reads and writes with it by configuring it using one of the above types (as long as the configured PLC is in the same PLC family). However, the READ STATUS command will not work since the type definition and the actual PLC definition will not match.

2.4.1.2 Editing Path Records

The following example shows the path records created by the configuration file:

```
create path SLC_PATH_4/vaxport="LTA994:"-
/netname=DH1/multidrop/timeout=3/log-
/line=(speed=9600,type,start=1,stop=1,data=8,parity=even,full,hangup)

create path DH1_PATH_5/vaxport="LTA995:"-
/netname=DH1/multidrop/timeout=3/log-
/line=(speed=9600,type,start=1,stop=1,data=8,parity=none,full,hangup)

create path DH1_PATH_7/vaxport="LTA997:"-
/netname=DH1/multidrop/timeout=3/log-
/line=(speed=9600,type,start=1,stop=1,data=8,parity=none,full,hangup)
```

The `/vaxport` field refers to the port (either LAT or TT) that is connected to the data highway line. If a direct connection to the device is made, then the path can be set as `POINT_TO_POINT`. If the connection is to a data highway network with more than one device configured, then the path must be set as `MULTIDROP`. The `/netname` field identifies the NI that is being used to communicate with the device.

Table 2–3 lists the path parameters and indicates the value(s) that are allowed for each.

Table 2–3 Data Highway Path Parameters

Parameter	Value(s)
VAXport[1]	LTAXXX:, TTXX:
Netname	DH1
Multidrop	MULTIDROP, POINT_TO_POINT
Timeout[2]	2-10
Retries[3]	
IO Size[3]	

[1]This parameter should be set to the OpenVMS device name used to connect to the data highway network or device.

[2]The timeout defaults to 3 seconds if no timeout is specified.

[3]The parameter is ignored by this DAS.

Table 2–4 lists the line parameters and indicates the value(s) that are allowed for each. If only one value is allowed, then the DAS will generate an error if the user sets the parameter to anything other than the allowed value. Verify that the VAXport and the line parameters match your plant-floor configuration.

Table 2–4 Line Parameters

Parameter	Value(s)
Stop Bits	1
Data Bits	8
Parity	even, none
Speed	300-19200
Full Duplex	full
Echo	noecho
Host Synch	hostsync, nohostsync
TT Synch	ttsync, nottsync
Typeahead	typeahead
Hangup	hangup
Modem	modem, nomodem
Dialup	dialup, nodialup
Disconnect	disconnect, nodisconnect

2.4.1.3 Editing Device Records

The following example shows the device records created by the configuration file:

```
create dev AB_1771_KE/path=DH1_PATH_5 /type=AB_PLC_2_IM /netaddr="22"/log

create dev AB_215/path=DH1_PATH_5/type=AB_PLC_2_15/timeout=120-
                /netaddr="15"/log

create dev AB_1770_KF2/path=DH1_PATH_7/type=AB_PLC_5_IM/netaddr="3"/log

create dev AB_5_250/path=DH1_PATH_7/type=AB_PLC_5_250/timeout=120-
                  /netaddr="1"/log

create dev AB_1771_KF2/path=DH1_PATH_5/type=AB_PLC_5_IM/netaddr="3"/log

create dev AB_5_10/path=DH1_PATH_5/type=AB_PLC_5_10/timeout=120-
                 /netaddr="1"/log

create dev SLC_5_04/path=DH1_PATH_5/type=AB_SLC_5_04/timeout=120-
                  /netaddr="4"/log
```

PLC-2, PLC-3 and PLC-5 models use octal addressing for the network address. SLC models use decimal addressing for the network address. The DAS assumes an octal address by default. To use a decimal address append a "D" to the network address (e.g. "12D")

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 the expected time of the longest device operation and also larger than the timeout on the path definition. To control the timeout for individual I/O operations to the device, use the timeout on the path definition.

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

2.4.2 DAS SPT Block Usage

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 2–5 shows the static blocks of each size that are used by the DAS.

Table 2–5 SPT Static Block Sizes

Block Size	Quantity[1]	Block Type
EXTRA LARGE	1	Device[3]
	1	Line
MEDIUM	1	Device
	10[2]	Device[3]
SMALL	1	Device
	1	DAS

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

[2]The number of blocks is 2 X the parameter ILAN$ABX_UNSOL_QUEUE_SIZE.

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

Table 2–6 shows the number of dynamic blocks of each size that are used by the DAS. These blocks are created and deleted as the device does I/O. I/O can be initiated solicitedly by a user or a polling set or can be initiated unsolicitedly by the device.

Table 2–6 SPT Dynamic Block Sizes

Block Size	Quantity[1]	I/O Type
LARGE	1	Upload[2], download[2]
MEDIUM	2	Read, write, status
	1	Upload[2], download[2]
	2	Upload[3], download[3]

[1]Quantity is quantity per I/O.

[2]PLC-5 only.

[3]Any PLC other than PLC-5.

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.

2.4.3 Configuring Ports

A device can be physically connected to a local area transport (LAT) port. To use a LAT port with the DAS, you must define the LAT port on the host system by using the LAT control program. The following DCL example shows how to define **port_2** on the LAT node **srvr4** as **LTA992**. It is very important when defining the LAT port to use the **/NOQUEUE** option. If the port is owned by another process device connection management reports an error during the establishment of a connection to the device. When you define a path and a device to BASEstar Classic device connection management LTA992 is supplied as the VAXport.

```
$ MCR LATCP

LCP> CREATE PORT LTA992

LCP> SET PORT LTA992 /NODE=srvr4/PORT=port_2/NOQUEUE
```

For the DAS for AB Data Highway Protocol to send and receive data correctly you must set and define LAT port characteristics to match the line parameters of the path definition for devices and terminals connected to the LAT. An example of the characteristics set at the LAT prompt follows:

```
Local> DEF PORT n SPEED 9600
Local> DEF PORT n AUTOBAUD DISABLED
Local> DEF PORT n AUTOCONNECT DISABLED
Local> DEF PORT n FLOW DISABLED
Local> DEF PORT n PREFERRED NONE
Local> DEF PORT n ACCESS REMOTE
Local> DEF PORT n CHAR 8
Local> DEF PORT n PARITY NONE
```

You must also set the following SYSGEN parameters to access Allen-Bradley PLCs to reduce the potential for data overruns. Use the SYSGEN utility to display the current values and AUTOGEN to reset the values accordingly. For information on using SYSGEN and AUTOGEN, refer to the *OpenVMS System Management Manual*.

Parameter	Value
TTY_ALTALARM	0
TTY_ALTYPAHD	1024

To enable use of the alternate typeahead buffer, type the following command from DCL:

```
$ SET TERM LTA992:/ALTYPEAHD/PERMANENT
```

The DAS will issue an error on a connection attempt if the ALTYPEAHD parameter has not been set.

2.4.4 Setting Up Plant-Floor Equipment

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

Table 2–7 provides switch settings to use in setting up your KE/KF/KF-2/KF-3 communications card.

Table 2–7 KE/KF/KF-2/KF-3 Switch Settings

Parameter	Supported	Values
Half Duplex	No	
Full Duplex	Yes	
Embedded Response	No	
BCC Error Check	Yes	
CRC Error Check	No	
RS-232C Handshaking	Yes[1]	
Station Number	Yes	1-377 (octal)[2], 1-31 (decimal)[3]
Computer Comm Rate	Yes	110 - 19200[4]

[1]If using the handshaking signals (RTS, CTS, DSR, DTR, DCD) be sure that the computer is also using those signals and that the cable has been wired properly.

[2]PLC, PLC-2, PLC-3, & PLC-5 controllers

[3]SLC controllers

[4]The 1771-KE, KF Series A Revision A-G support a maximum of 9600 baud. The 1771-KE, KF Series A Revision H and later, the 1770-KF2 Series B Revision F and later and the 1785-KE support a maximum of 19200 baud.

2.5 Tracing Device Communications

The DAS for AB Data Highway Protocol has built into it the capability to trace PLC communications and to send this output to either a file or a user terminal. The purpose of this tracing is to allow the troubleshooting of PLC communications.

To enable tracing do the following:

- Set the following logical to 1:

  DCM_AB$DH1_TRACE Trace the NI (Network Interface)

  The logical name must be defined in a scope that will be seen by the device server process (ILAN$DEVSRV).

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

  DCM_AB$DH1_TRACE_OUTPUT Trace the NI (Network Interface)

  The logical name must be defined in a scope that will be seen by the device server process (ILAN$DEVSRV).

- Start the tracing.

  To start tracing for the NI, disable all devices on the line for which tracing is desired and reenable them. The trace logical is read when the first device on the line is enabled.

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.

2.6 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 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 PLCs, and how to access these functions.

3.1 Accessing DAS Functions

The DAS for AB Data Highway Protocol functions are accessed through the BASEstar Classic device connection management:

- Commands
- Menu system
- Callable services

To use the BASEstar Classic device connection management commands, enter the following command at the DCL prompt ($):

```
$  BSTAR DCM
```

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

To use the BASEstar Classic device connection management menu system, enter the following command:

```
$  BSTAR/MENU
```

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

Note

The menu system is available on OpenVMS/VAX systems only.

For additional information about the BASEstar Classic device connection management callable services, refer to the *BASEstar Classic Introduction to Callable Services*.

3.2 Supported Functions

The following sections describe the functions that are supported by the DAS for AB Data Highway Protocol. Table 3–1 displays the Allen-Bradley PLCs and the functions these devices support.

Table 3–1 Allen-Bradley Devices and Functions

Device	Upload	Download	Start	Stop	Read Data	Write Data	Read Status
PLC-1774	X	X	X	X	X	X	X
PLC-2 family	X	X			X	X	X
PLC-3 family	X	X	X	X	X	X	X
PLC-5 family	X	X	X	X	X	X	X
PLC-5/250			X	X	X	X	X
SLC family					X	X	X

3.2.1 Start and Stop

The start and stop functions change the operating mode of a device. The start function enables outputs and the stop function disables outputs. These functions are not supported for the PLC-2 family of devices.

Warning

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

3.2.1.1 PLC-1774

The start function enables outputs and scanning on the PLC-1774 device. Using the STOP command when the keyswitch position is RUN/PROGRAM LOAD - RUN causes the status of the device to appear as Remote Test. A subsequent START command enables outputs.

Table 3–2 displays the keyswitch positions, current mode, request types, DAS activity, and resulting mode for the PLC-1774 while performing the start and stop functions.

Table 3–2 Start and Stop Function

Keyswitch Position	Current Mode	Function	DAS Activity	Resulting Mode
PROGRAM LOAD	Program load	Start	Invalid mode	Program load
	Program load	Stop	None	Program load
TEST MONITOR	Test mode	Start	Invalid mode	Test
	Test mode	Stop	None	Test
RUN MONITOR	Run mode	Start	Enable outputs	Run
	Run mode	Stop	Disable outputs	Run
RUN/PROGRAM LOAD - RUN	Remote run	Start	Enable scanning,Enable outputs	Remote run
	Remote run	Stop	Disable outputs	Remote test
RUN/PROGRAM LOAD - TEST	Remote test	Start	Enable scanning	Remote test
	Remote test	Stop	None	Remote test
RUN/PROGRAM LOAD - PROG	Remote prog	Start	Invalid mode	Remote prog
	Remote prog	Stop	None	Remote prog

Note

PLC-1774 outputs are disabled with a STOP command, but scanning is not stopped.

3.2.1.2 PLC-3

The start function changes the current mode to Run. The stop function changes the current mode to Program Load.

3.2.1.3 PLC-5

The Start function changes the device operating mode to Remote Run. The Stop function changes the device operating mode to Remote Program. The device keyswitch must be in the REM position for the start or stop request to change the device's operating mode.

3.2.1.4 SLC

Start and Stop functions are not supported for SLC processores.

3.2.2 Upload and Download

The upload function transfers the contents of a device's memory to an OpenVMS file. You can specify the STOP qualifier when using the UPLOAD command. If you specify the STOP qualifier, the PE prevents the contents of PLC memory from being modified during the upload. All BASEstar Classic device connection management write functions are rejected with a device busy message.

The download function transfers the contents of an OpenVMS file to a device's memory. You can specify the STOP and RESTART qualifiers to control the device mode when using the DOWNLOAD command. The PE does not allow the operating status of a PLC to be changed unless specified in the DOWNLOAD command.

You must specify the STOP qualifier when you download a running device. The STOP qualifier is a safety mechanism. You must specify the RESTART qualifier to start the device when the download is completed. All device connection management read (except read status) and write functions are rejected with a device busy message during the download.

3.2.2.1 PLC-1774

When you perform the upload function, the keyswitch must be in the RUN-MONITOR, RUN/PROGRAM LOAD, or TEST-MONITOR position. If the keyswitch is in the RUN/PROGRAM LOAD position, the PLC programming terminal must have the device mode set to Remote RUN-MONITOR or TEST-MONITOR. The STOP qualifier for a PLC-1774 upload function prevents other processes from writing to the device through the DAS. The syntax for the STOP qualifier follows:

```
UPLOAD dev_name filename /QUALIFIER="STOP"
```

When you perform the download function, the keyswitch must be in the RUN-MONITOR, RUN/PROGRAM LOAD, or TEST-MONITOR position. If the keyswitch is in the RUN/PROGRAM LOAD position, the PLC programming terminal must have the device mode set to Remote RUN-MONITOR or Remote TEST-MONITOR. There are two qualifiers associated with the download operation:

- STOP

  The STOP qualifier for a PLC-1774 allows the device to be downloaded while running. The PLC-1774 cannot be downloaded without the STOP qualifier because it must be scanning when the download is initiated.

- RESTART

  The RESTART qualifier sends both scan enable and enable outputs commands to the device when the download completes.

The syntax for the STOP and RESTART qualifiers follows:

```
DOWNLOAD dev_name filename /QUALIFIER="STOP/RESTART"
```

Note

You cannot download a PLC-1774 without stopping the device. STOP is a mandatory qualifier.

Table 3–3 displays the modes, qualifiers, and supported functions when you use the download function:

Table 3–3 PLC-1774 Download Function

			After Download	
Keyswitch Position	**Qualifiers**	**Scan**	**Outputs**	**Mode**
RUN MONITOR	Stop	Disable outputs	Disable outputs	RUN
	Stop/Restart	Enable outputs	Enable outputs	RUN
RUN/PROGRAM LOAD - RUN	Stop	Disable outputs	Disable outputs	Remote test
	Stop/Restart	Enable outputs	Enable outputs	Remote run
TEST MONITOR	Stop	Disable outputs	Not modified	TEST
	Stop/Restart	Enable outputs	Not modified	TEST
RUN/PROGRAM LOAD - TEST	Stop	Disable outputs	Not modified	Remote test
	Stop/Restart	Enable outputs	Not modified	Remote test, remote run

3.2.2.2 PLC-2

You can perform the upload function for the PLC-2 devices in any mode. If you issue the UPLOAD command with the STOP qualifier, the device is issued an upload request. If you are using a 1771-KA interface card and issue the UPLOAD command with the STOP qualifier, the device is issued a download request. The STOP qualifier disables the industrial terminal as well as preventing other processes from writing to the device through the DAS.

Table 3–4 displays the interface, modes, qualifiers, and functions supported, modified, and not supported by the upload function.

Table 3–4 Upload Function

Interface	Mode	Qualifier	Read	Write
1771-KA	Prog	None	Not supported	Not supported
	Remote prog	None	Not supported	Not supported
	Prog	Stop	Not supported	Not supported
	Remote prog	Stop	Not supported	Not supported
	Run	None	Supported	Supported
	Remote run	None	Supported	Supported
	Test	None	Supported	Supported
	Run test	None	Supported	Supported
	Run	Stop	Supported	Not supported
	Remote run	Stop	Supported	Not supported
	Test	Stop	Supported	Not supported
	Run test	Stop	Supported	Not supported
1771-KA2	Prog	None	Not supported	Not supported
	Remote prog	None	Not supported	Not supported
	Prog	Stop	Not supported	Not supported
	Remote prog	Stop	Not supported	Not supported
	Run	None	Supported	Supported
	Remote run	None	Supported	Supported
	Test	None	Supported	Supported
	Run test	None	Supported	Supported
	Run	Stop	Supported	Not supported
	Remote run	Stop	Supported	Not supported
	Test	Stop	Supported	Not supported
	Run test	Stop	Supported	Not supported

Note

Uploading with the STOP qualifier disables the industrial terminal until the upload function is completed.

The action taken for the download function (and associated qualifiers) for a PLC-2 family device depends on the communications interface card being used with the device. Certain revisions of the 1771-KA2 communications adapter do not allow a logic program to be downloaded to the memory of a PLC-2/20 in the RUN/PROG mode.

To determine whether or not the combination of a given revision of the 1771-KA2 interface and PLC-2/20 processor produces this problem, contact an Allen-Bradley representative. Until Allen-Bradley releases a revision of hardware that allows

downloading in the RUN/PROG mode, some PLC-2/20s may only be downloaded in the PROG mode.

The 1771-KA interface does not support the SET DATA TABLE SIZE command. The file data table downloaded must have the same data table size as the destination device. If the download function fails with an invalid data table size error, check the history file for the file data table size and reset the device data table size using an industrial terminal. Using the download function through a 1771-KA2 interface sets the device data table size to match the download file data table size.

Valid download functions for the 1771-KA1 and 1771-KA2 interfaces are:

- TEST or RUN/PROG Test mode - perform download function with the STOP and RESTART qualifiers
- PROG or RUN/PROG Prog mode - perform download function with no qualifier or the STOP qualifier
- RUN or RUN/PROG Run mode - perform download function with the STOP and RESTART qualifiers

Note

The RUN and RUN/PROG Run mode are not supported through the 1771-KA interface during the download function. The RUN and TEST modes are not supported through the 1771-KA2 interface during the download function. The read and write functions are not supported during a download function on a PLC-2 device.

You must specify the STOP qualifier if the device is to be stopped by the download function. Use the RESTART qualifier to restart the device after the download function is completed.

3.2.2.3 PLC-3

You can perform the upload function for the PLC-3 devices in any mode. The STOP qualifier prevents other processes from writing to the device through the DAS. PLC-3 file allocations are frozen during the upload function regardless of the STOP qualifier.

The controller performs the upload function until the end of the program is reached. Following the upload function, the PLC-3 is returned to its original state.

You can perform the download function for a PLC-3 family device in any mode. If the device is in Run or Test mode, the STOP qualifier is *mandatory*. If you specify the RESTART qualifier, the device is put in Run mode when the download function completes. If you omit the RESTART qualifier, the device is put in Program mode when the download function completes. The PLC-3's station address is preserved and the station address in the logic file is ignored.

To perform the download function, set the keyswitch to the MEMORY PROTECT OFF position.

3.2.2.4 PLC-5

The upload function can be performed by a PLC-5/XX device in any mode. Upload is not supported for the PLC-5/250.

Specifying /QUALIFIER=STOP helps to prevent modification of device memory during an upload. If you specify the STOP qualifier, the device cannot be scanning during the upload function. If the device is in Remote Run or Remote Test and you specify the STOP qualifier, the device mode is changed to Remote Program until the upload function completes. Note that the PLC-5/15 (Series B, Revision E, or earlier) does not support any read functions except read status during an upload function when you specify the STOP qualifier.

The keyswitch must be in the REMOTE position to use the following qualifiers:

- STOP - puts the device in REM PROG mode before performing the upload function
- RESTART - puts the device in REM RUN mode after performing the upload function

You cannot restart in PROG mode or stop in RUN mode while performing an upload function. The STOP qualifier prevents other processes from writing to the device through the DAS for AB Data Highway Protocol during the upload.

When you perform the download function for a PLC-5 family device, the device can be in any mode except RUN mode. The keyswitch position must be REM or PROG. Note that if the device is in Remote Run or Remote Test mode, the STOP qualifier is *mandatory*. The RESTART qualifier is only valid if the keyswitch is in the REM position. If you specify the RESTART qualifier, the device is put in the Remote Run mode when the download function completes.

The download function supports the following qualifiers:

- STOP - puts the device in REM PROG mode before performing the download function
- RESTART - puts the device in REM RUN mode after performing the download function
- OVERRIDE - allows a file from one PLC-5 model to be downloaded to a different PLC-5 model. Without this qualifier, a file may only be downloaded to the same model it was uploaded from.

You must specify the STOP qualifier if the device is to be stopped by the download function.

Table 3–5 displays the combinations of modes and qualifiers available while performing a download operation. An "N" indicates an invalid combination and a "Y" indicates a valid combination.

Table 3–5 PLC-5 Download Function

	Qualifiers			
Mode	**None**	**Stop**	**Restart**	**Stop/Restart**
RUN	N	N	N	N
PROG	Y	Y	N	N
REM RUN	N	Y	N	Y
REM PROG	Y	Y	Y	Y
REM TEST	N	Y	N	Y

3.2.2.5 SLC

Upload and Download functions are not supported for SLC processors.

3.2.3 Structured Data

The DAS for AB Data Highway Protocol supports the use of structured data. Structured data is a block of data, typically represented by a buffer, that can be broken down into meaningful elements of data. Structured data allows you to impose a structure on the data formats for any given read or write operation. A structure tells the device how to interpret the data that it is reading or writing.

For example, temperature and humidity data could be packed into a longword and a word in a buffer. The buffer is then said to contain structured data, with the longword representing temperature, and the word representing humidity.

For the DAS for AB Data Highway Protocol, all data types, except bit strings, are supported for structured data when using PLC-2, PLC-3 or PLC-5 models. When using SLC models all elements in a structure must be of the same format.

3.2.4 Read and Write Data

Use the read data and write data functions to read data from and write data to a specific address in device storage. Address syntax is specific to each device or device family, and is similar to the syntax used with the Allen-Bradley T3, T4, or T50 programming terminals. The data type formats supported depend on the device.

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

Example 3–1 Read Data

```
DCM> read data ab_5/format=array[16]:bit/address="N7:00/00"

Device: AB_5
Data starting at address N7:00/00 for ARRAY[16]:BIT element.
Data:
   0:    1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

Example 3–2 shows an example of the information displayed when you enter the READ DATA command with structured data.

Example 3–2 READ DATA With Structured Data

```
DCM> read data ab_3/addr=I300/form="struct(word,word,word)

Device      : AB_3
Address     : I300
Format      : STRUCTURE(S_WORD,S_WORD,S_WORD)

Data:
    0       :        1
    1       :        2
    2       :        3
```

Example 3–3 WRITE DATA

```
DCM> write data ab_5/format=word/address=N7:00
Data Value         1 : 3
```

Example 3–4 WRITE DATA with Structured Data

```
DCM> WRITE DATA AB_3/addr=I300/form="struct(word,word,word)

Device      : AB_3
Address     : I300
Format      : STRUCTURE(S_WORD,S_WORD,S_WORD)

Data value     0       :        0
Data value     1       :        0
Data value     2       :        0
```

Example 3–3 shows an example of the information displayed when you enter the WRITE DATA command.

Example 3–4 shows an example of the information displayed when you enter the WRITE DATA command with structured data.

The following sections describe the use of the read data and write data functions by specific devices, including addressing and data type formats.

3.2.4.1 PLC-1774

Addressing and data type formats are necessary for performing the read data and write data functions for the PLC-1774 device.

Addressing

Legal address formats for the PLC-1774 are:

www
wwww
wwwbb

where:

www = 0-777 octal word address
wwww = 0-1777 octal word address
www = 0-377 octal word
bb = 0-17 octal bit

The following examples are legal addresses for the PLC-1774:

```
400          (word 400)
1600         (word 1600)
36610        (word 366, bit 10)
```

Data Type Formats

Legal data type formats for the PLC-1774 are bit, word, unsigned word, byte, unsigned byte, and packed decimal. Reading data from the device into the format of word, unsigned word, or packed decimal eliminates the status bits that are present in the internal binary coded decimal format of the device. Writing data to the device from the format of word, unsigned word, or packed decimal is limited to 0-999, and the resulting status bits are cleared.

3.2.4.2 PLC-2

Addressing and data type formats are necessary for performing the read data and write data functions for the PLC-2 family of devices.

Addressing

The address format for the PLC-2 family is:

wwwww/bb

where:

wwwww = 0-17777 octal word address
bb = 0-17 octal bit (optional)

The maximum allowable word address differs for each member of the PLC-2 family.

The following examples are legal addresses for the PLC-2:

```
20        (word 20)
372/00    (word 372, bit 00)
17777/17  (word 17777, bit 17)
```

Data Type Formats

Legal data type formats for the PLC-2 family are bit, word, unsigned word, byte, unsigned byte, and packed decimal. Reading data from the device into the format of word, unsigned word, or packed decimal eliminates the status bits that are present in the internal BCD format of the device. Data to be written to the device from the format of word, unsigned word, or packed decimal is limited to 0-999, and the resulting status bits are cleared.

3.2.4.3 PLC-3

Addressing and data type formats are necessary for performing the read data and write data functions for the PLC-3 family of devices.

Addressing

There are two types of addressing for PLC-3 devices: extended addressing and data table addressing. Extended addressing has the general format:

E.M.S.S.S.S.W/B

where:

E is the extended address indicator
M is the major area number
S is the subarea designation
W is the word address

B is the bit address

There can be as little as two subarea designations and a maximum of four, depending on which major area is being addressed.

Data table addressing simplifies the addressing of the data table, and generally has the format:

ASF:W/B

where:

A is the basic structure: W for word (default), F for file
S is the section of the data table: O, I, T, C, N, F, D, B, A, H, P, S
F is the file number: 0 - 999 (default is the file 0)
W is the word number within the file: 0 - 999; 0 - 7777 octal for I/O
B is the bit: 0 - 17 octal within word; 0 - 9999 within file

You can access data within each data table section at file, word, or bit levels.

Bit numbers are octal for addresses having word structures and decimal for addresses having file structures. For example, you can address the same bit as WN2:4/10, FN2:4/8, or FN2:0/72. In the first address (WN2:4/10), W indicates word structure for integer section data stored in file 2, word 4, bit 10 (octal). The last two addresses (FN2:4/8 and FN:0/72) use file addressing. In these examples, F indicates file structure. To locate the bit with the address FN2:4/8, the processor counts 8 bits in integer file 2 starting with word 4, bit 0. With the address FN2:0/72, the processor counts 72 bits in integer file 2 starting with word 0, bit 0.

Certain data table sections may be addressed differently; refer to the *PLC-3 Programmable Controller Programming Manual* for details.

The following examples are legal addresses for the PLC-3:

```
I370             (input section, rack 37, module 0)
FN2:738/17       (integers section, file 2, word 738, bit 17)
TACC9990         (timers section, accumulator word, structure 9990)
E3.1.9.0.0.1010  (ASCII section (9), word 1010)
E3.1.4.0.13.2/7  (counters section (4), structure 13, word 2, bit 7)
```

Data Type Formats

Table 3–6 shows the supported data type formats for reading from and writing to the various data table sections of the PLC-3.

Table 3–6 Read and Write Data Functions

	Data Type Formats						
PLC-3 Data Section	**Bit**	**Word**	**Unsigned Word**	**Byte**	**Unsigned Byte**	**Floating Point**	**Longword**
Output	X	X	X	X	X		
Input	X	X	X	X	X		
Timers	X	X	X	X	X		
Counters	X	X	X	X	X		
Integers	X	X	X	X	X		
Floating Point	X			X	X	X	
Decimal	X	X	X	X	X		
Binary	X	X	X	X	X		
ASCII	X			X	X		
High Order Ints	X			X	X		X
Pointers	X	X	X	X	X		X
Status	X	X	X	X	X		

3.2.4.4 PLC-5

Address syntax and supported device connection management data types for read and write operations vary between the PLC-5/250 and the other PLC-5 family devices.

The following sections describe the use of the read data and write data functions by specific devices, including addressing and data type formats. Addressing and data type formats are necessary for performing the read data and write data functions for the PLC-5 family of devices.

Data Type Formats

Table 3–7 shows which data type formats are supported for reading from and writing to the various file types of the PLC-5 family.

Table 3–7 PLC-5 Data Format General Reference

Device File Type	Bit	Byte	Word	Long	Float	String
Output	X		X			
Input	X		X			
Status	X		X			
Binary	X		X			
Timer	X		X	X		
Counter	X		X			
Control	X		X			
Integer	X		X			
Floating Point					X	
ASCII	X	X				X
Long Integer	X			X		
PID Control	X	X			X	
Internal Storage	X		X			
Shared Data	X		X			
Block Transfer Data	X		X			
String		X	X			X
Adapter Status	X	X	X			
Message	X	X	X			

General Address Formats

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

```
mFSf:nn/bb  or  mFSf:nn.mne
```

where:

- m = module number (module numbers are 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

Specific Address Formats

The following tables describe address specifications and data types for specific PLC-5 family file types. Refer to the general address formats listed above for a description of the conventions used to describe address specifications.

Input File Addresses

Table 3–8 describes address specifications and data types for input file addresses. Input file addresses are identical for all PLC-5 devices and are specified in octal.

Table 3–8 Input File Addresses

Address	Format	Comment
I :rr	Word	Input image table word
I :rr/bb	Bit	Output image table bit

Output File Addresses

Table 3–9 describes address specifications and data types for output file addresses. Output file addresses are identical for all PLC-5 devices and are specified in octal.

Table 3–9 Output File Addresses

Address	Format	Comment
O :rr	Word	Output image table word
O :rr/bb	Bit	Output image table bit

Status File Addresses

Table 3–10 describes address specifications and data types for status file addresses. Status file addresses are identical for all PLC-5 devices, except that a module number must be specified for the PLC-5/250.

Table 3–10 Status File Addresses

Address	Format	Comment
mS:nn	Word	Status file word
mS:nn/bb	Bit	Status file bit

Binary File Addresses

Table 3–11 describes address specifications and data types for binary file addresses. Binary file addresses are identical for all PLC-5 devices, except that a module number must be specified for the PLC-5/250.

Table 3–11 Binary File Addresses

Address	Format	Comment
mBf:nn	Word	Binary file word
mBf:nn/bb	Bit	Binary file bit

Timer File Addresses

Table 3–12 describes address specifications and data types for timer file addresses. Timer file addresses are identical for all PLC-5 devices, except that a module number must be specified for the PLC-5/250. Timers can be read as entire structures or addressed at the sub-element level.

For a PLC-5/250, the accumulator and preset values are longwords and the total structure size is 12. For other PLC-5 family devices, the accumulator and preset values are words.

Table 3–12 Timer File Addresses

Address	Format	Comment
mTf:nn	Array[3]:long	PLC-5/250 timer structure
Tf:nn	Array[3]:word	PLC-5/[12,15,25] timer structure
mTf:PRE	Long	PLC-5/250 timer preset
Tf:PRE	Word	PLC-5/[12,15,25] timer preset
mTf:PRE/bb	Bit	Timer preset bit
mTf:ACC	Long	PLC-5/250 timer accumulator
Tf:ACC	Word	PLC-5/[12,15,25] timer accumulator
mTf:ACC/bb	Bit	Timer accumulator bit
mTf:EN[1]	Bit	Timer enable bit
mTf:TT[1]	Bit	Timer timing bit
mTf:DN[1]	Bit	Timer done bit

[1] Control bits should be treated as read only and modified only under PLC program control. Modifying these bits from a computer can give unpredictable results.

Counter File Addresses

Table 3–13 describes address specifications and data types for counter file addresses. Counter file addresses are identical for all PLC-5 devices except, that a module number must be specified for the PLC-5/250. Counters can be read as entire structures or addressed at the sub-element level.

Table 3–13 Counter File Addresses

Address	Format	Comment
mCf:nn	Array[3]:word	Entire counter structure
mCf:PRE	Word	Counter preset
mCf:PRE/bb	Bit	Counter preset bit
mCf:ACC	Word	Counter accumulator
mCf:ACC/bb	Bit	Counter accumulator bit
mCf:CU[1]	Bit	Counter up enable bit
mCf:CD[1]	Bit	Counter down enable bit
mCf:DN[1]	Bit	Counter done bit
mCf:OV[1]	Bit	Counter overflow bit
mCf:UN[1]	Bit	Counter underflow bit

[1] Control bits should be treated as read only and modified only under PLC program control. Modifying these bits from a computer can give unpredictable results.

Control File Addresses

Table 3–14 describes address specifications and data types for control file addresses. Control file addresses are identical for all PLC-5 devices, except that a module number must be specified for the PLC-5/250.

Table 3–14 Control File Addresses

Address	Format	Comment
mRf:nn	Array[3]:word	Entire control structure
mRf:nn.LEN	Word	Control expression files length
mRf:nn.LEN/bb	Bit	Length bit
mRf:nn.POS	Word	Control expression file position
mRf:nn.POS/bb	Bit	Position bit
mRf:nn.EN[1]	Bit	Enabled bit
mRf:nn.EU[1]	Bit	Enabled unloading bit
mRf:nn.DN[1]	Bit	Done bit
mRf:nn.EM[1]	Bit	Empty bit
mRf:nn.ER[1]	Bit	Error bit
mRf:nn.UL[1]	Bit	Unload bit
mRf:nn.IN[1]	Bit	Inhibit comparisons bit
mRf:nn.FD[1]	Bit	Found bit

[1] Control bits should be treated as read only and modified only under PLC program control. Modifying these bits from a computer can give unpredictable results.

Integer File Addresses

Table 3–15 describes address specifications and data types for integer file addresses. Integer file addresses are identical for all PLC-5 devices, except that a module number must be specified for the PLC-5/250.

Table 3–15 Integer File Addresses

Address	Format	Comment
mNf:nn	Word	Integer file word
mNf:nn/bb	Bit	Integer file bit

Floating Point File Addresses

Table 3–16 describes address specifications and data types for floating point file addresses. Floating point file addresses are identical for all PLC-5 devices, except that a module number must be specified for the PLC-5/250.

Table 3–16 Floating Point File Addresses

Address	Format	Comment
mFf:nn	Float	Float file floating point

ASCII File Addresses

Table 3–17 describes address specifications and data types for ASCII file addresses. ASCII file addresses are supported only for the PLC-5/10, PLC-5 /11, PLC-5/12, PLC-5/15, PLC-5/20, PLC-5/25, PLC-5/30, PLC-5/40, PLC-5/60 and the PLC-5/80. ASCII addresses must be read from and written to with element counts that are multiples of 2.

Table 3–17 ASCII File Addresses

Address	Format	Comment
Af:nn	Array[2]:byte	2 ASCII file characters

Long Integer File Addresses

Table 3–18 describes address specifications and data types for long integer file addresses. Long integer file addresses are supported only for the PLC-5/250.

Table 3–18 Long Integer File Addresses

Address	Format	Comment
mLf:nn	Word	Long integer file longword
mNf:nn/bb	Bit	Long integer file bit

PID Control File Addresses

Table 3–19 describes address specifications and data types for PID control file addresses. PID control file addresses are supported only for the PLC-5/250.

Table 3–19 PID Control File Addresses

Address	Format	Comment
mPDf:nn	Array[164]:byte	Entire PID control structure
mPDf:nn/bb	Bit	PID control structure bit
mPDf:nn.SP	Float	Setpoint
mPDf:nn.KP	Float	Proportional gain
mPDf:nn.KI	Float	Integral gain
mPDf:nn.KD	Float	Derivative time
mPDf:nn.BIAS	Float	Output bias %
mPDf:nn.MAXS	Float	Setpoint maximum
mPDf:nn.MINS	Float	Setpoint minimum
mPDf:nn.DB	Float	Deadband
mPDf:nn.SO	Float	Set output %
mPDf:nn.MAXO	Float	Output limit high %
mPDf:nn.MINO	Float	Output limit low %
mPDf:nn.UPD	Float	Update time
mPDf:nn.PV	Float	Process variable
mPDf:nn.ERR	Float	Error
mPDf:nn.OUT	Float	Output
mPDf:nn.PVH	Float	PV alarm high
mPDf:nn.PVL	Float	PV alarm low
mPDf:nn.DVP	Float	Deviation alarm +
mPDf:nn.DVN	Float	Deviation alarm -

(continued on next page)

Table 3–19 (Cont.) PID Control File Addresses

Address	Format	Comment
mPDf:nn.PVDB	Float	PV alarm deadband
mPDf:nn.MAXI	Float	Input range maximum
mPDf:nn.MINI	Float	Input range minimum
mPDf:nn.TIE	Float	Tieback %
mPDf:nn.ADDR	Float	Address of master loop
mPDf:nn.DVDB	Float	Deviation alarm deadband
mPDf:nn.DATA	Array[14]:float	Reserved - interim use
mPDf:nn.EN[1]	Bit	PID enable bit
mPDf:nn.CT[1]	Bit	Cascaded type bit
mPDf:nn.CL[1]	Bit	Cascaded loop bit
mPDf:nn.PVT[1]	Bit	PV tracking bit
mPDf:nn.DO[1]	Bit	Derivative off bit
mPDf:nn.SWM[1]	Bit	Software A/M mode bit
mPDf:nn.CA[1]	Bit	Control action bit
mPDf:nn.MO[1]	Bit	Mode bit
mPDf:nn.PE[1]	Bit	PID equation bit
mPDf:nn.INI[1]	Bit	PID initialized bit
mPDf:nn.SPOR[1]	Bit	SP out of range bit
mPDf:nn.OLL[1]	Bit	Output limit low bit
mPDf:nn.OLH[1]	Bit	Output limit high bit
mPDf:nn.EWD[1]	Bit	Error within deadband bit
mPDf:nn.DVNA[1]	Bit	Deviation high alarm bit
mPDf:nn.DVPA[1]	Bit	Deviation low alarm bit
mPDf:nn.PVLA[1]	Bit	PV low alarm bit
mPDf:nn.PVHA[1]	Bit	PV high alarm bit

[1] Control bits should be treated as read only and modified only under PLC program control. Modifying these bits from a computer can give unpredictable results.

Internal Storage File Addresses

Table 3–20 describes address specifications and data types for internal storage file addresses. Internal storage file addresses are supported only for the PLC-5/250 and are specified in octal.

Table 3–20 Internal Storage File Addresses

Address	Format	Comment
IS:rr	Word	Internal storage file word
IS:rr/bb	Bit	Internal storage file bit

Shared Data File Addresses

Table 3–21 describes address specifications and data types for shared data file addresses. Shared data file addresses are supported only for the PLC-5/250.

Table 3–21 Shared Data File Addresses

Address	Format	Comment
mSDf:nn	Word	Shared data file word
mSDf:nn/bb	Bit	Shared bit

Block Transfer Data Addresses

Table 3–22 describes address specifications and data types for block transfer data addresses. Block transfer data addresses are supported only for the PLC-5/250.

Table 3–22 Block Transfer Data Addresses

Address	Format	Comment
mBTDf:nn	Word	Block transfer data file word
mBTDf:nn/bb	Bit	Block transfer data bit

String Data Addresses

Table 3–23 describes address specifications and data types for string data addresses. String data addresses are supported only for the PLC-5/250.

Table 3–23 String Data Addresses

Address	Format	Comment
mSTf:nn	String	Entire string structure
mSTf:nn.LEN	Word	String length
mSTf:nn.DATA	Array[82]:byte	String data—up to 82 bytes

Adapter Status File Addresses

Table 3–24 describes address specifications and data types for adapter status file addresses. Adapter status file addresses are supported only for the PLC-5/250 and are specified in octal.

Table 3–24 Adapter Status File Addresses

Address	Format	Comment
ASf:rr	Array[4]:byte	Entire adapter status structure
ASf:rr.RC	Byte	Adapter status retry count
ASf:rr.OI[1]	Bit	Adapter status output inhibit
ASf:rr.CF[1]	Bit	Adapter status fault

[1] Control bits should be treated as read only and modified only under PLC program control. Modifying these bits from a computer can give unpredictable results.

Message Addresses

Table 3–25 describes address specifications and data types for message addresses. Message addresses are supported only for the PLC-5/250.

Table 3–25 Message Addresses

Address	Format	Comment
mMSGf:nn	Array[112]:byte	Entire message structure
mMSGf:nn.ERR	Word	Message error code
mMSGf:nn.ERR/bb	Bit	Message error code bit
mMSGf:nn.RLEN	Word	Message request length
mMSGf:nn.RLEN /bb	Bit	Message request length bit
mMSGf:nn.DLEN	Word	Message done length
mMSGf:nn.DLEN /bb	Bit	Message done length bit
mMSGf:nn.EN[1]	Bit	Message enable bit
mMSGf:nn.ST[1]	Bit	Message start transmission bit
mMSGf:nn.AD[1]	Bit	Message done bit
mMSGf:nn.AE[1]	Bit	Message error bit
mMSGf:nn.CO[1]	Bit	Message continuous bit
mMSGf:nn.EW[1]	Bit	Message enabled waiting
mMSGf:nn.DN[1]	Bit	Message synchronization done
mMSGf:nn.ER[1]	Bit	Message synchronization error

[1] Control bits should be treated as read only and modified only under PLC program control. Modifying these bits from a computer can give unpredictable results.

3.2.4.5 SLC

Address syntax and supported device connection management data types for read and write operations vary between the SLC-5/04 and SLC-5/03 and other models of the SLC family of devices.

The following sections describe the use of the read data and write data functions by specific devices, including addressing and data type formats. Addressing and data type formats are necessary for performing the read data and write data functions for the SLC family of devices.

Data Type Formats

Table 3–26 shows which data type formats are supported for reading from and writing to the various file types of the SLC family.

Table 3–26 SLC Data Format General Reference

Device File Type	Bit	Word	Long	Float	String
Status	X	X			
Binary	X	X			
Timer	X	X			
Counter	X	X			
Control	X	X			
Integer	X	X	X		X
Floating Point				X	
ASCII					X
String					X

Note

ASCII, String and Floating Point are available only on SLC 5/03 Series C with OS301 and SLC 5/04 Series A with OS400.

General Address Formats

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 = bit address (decimal)
- mne = sub-element or bit mnemonic

Specific Address Formats

The following tables describe address specifications and data types for specific SLC family file types. Refer to the general address formats listed above for a description of the conventions used to describe address specifications.

Status File Addresses

Table 3–27 describes address specifications and data types for status file addresses.

Table 3–27 Status File Addresses

Address	Format	Comment
mS:nn	Word	Status file word
mS:nn/bb	Bit	Status file bit

Binary File Addresses

Table 3–28 describes address specifications and data types for binary file addresses.

Table 3–28 Binary File Addresses

Address	Format	Comment
mBf:nn	Word	Binary file word
mBf:nn/bb	Bit	Binary file bit

Timer File Addresses

Table 3–29 describes address specifications and data types for timer file addresses. Timers can be read as entire structures or addressed at the sub-element level.

Table 3–29 Timer File Addresses

Address	Format	Comment
Tf:nn	Struct(Array[3]:word)	Timer structure
Tf:PRE	Word	Timer preset
mTf:PRE/bb	Bit	Timer preset bit
Tf:ACC	Word	Timer accumulator
mTf:ACC/bb	Bit	Timer accumulator bit
mTf:EN[1]	Bit	Timer enable bit
mTf:TT[1]	Bit	Timer timing bit
mTf:DN[1]	Bit	Timer done bit

[1] Control bits should be treated as read only and modified only under PLC program control. Modifying these bits from a computer can give unpredictable results.

Counter File Addresses

Table 3–30 describes address specifications and data types for counter file addresses. Counters can be read as entire structures or addressed at the sub-element level.

Table 3–30 Counter File Addresses

Address	Format	Comment
mCf:nn	Struct(Array[3]:word)	Entire counter structure
mCf:PRE	Word	Counter preset
mCf:PRE/bb	Bit	Counter preset bit
mCf:ACC	Word	Counter accumulator
mCf:ACC/bb	Bit	Counter accumulator bit
mCf:CU[1]	Bit	Counter up enable bit
mCf:CD[1]	Bit	Counter down enable bit
mCf:DN[1]	Bit	Counter done bit
mCf:OV[1]	Bit	Counter overflow bit

[1] Control bits should be treated as read only and modified only under PLC program control. Modifying these bits from a computer can give unpredictable results.

(continued on next page)

Table 3–30 (Cont.) Counter File Addresses

Address	Format	Comment
mCf:UN[1]	Bit	Counter underflow bit
mCf:UA[1]	Bit	Counter update accumulator bit

[1] Control bits should be treated as read only and modified only under PLC program control. Modifying these bits from a computer can give unpredictable results.

Control File Addresses

Table 3–31 describes address specifications and data types for control file addresses.

Table 3–31 Control File Addresses

Address	Format	Comment
mRf:nn	Struct(Array[3]:word)	Entire control structure
mRf:nn.LEN	Word	Control expression files length
mRf:nn.LEN/bb	Bit	Length bit
mRf:nn.POS	Word	Control expression file position
mRf:nn.POS/bb	Bit	Position bit
mRf:nn.EN[1]	Bit	Enabled bit
mRf:nn.EU[1]	Bit	Enabled unloading bit
mRf:nn.DN[1]	Bit	Done bit
mRf:nn.EM[1]	Bit	Empty bit
mRf:nn.ER[1]	Bit	Error bit
mRf:nn.UL[1]	Bit	Unload bit
mRf:nn.IN[1]	Bit	Inhibit comparisons bit
mRf:nn.FD[1]	Bit	Found bit

[1] Control bits should be treated as read only and modified only under PLC program control. Modifying these bits from a computer can give unpredictable results.

Integer File Addresses

Table 3–32 describes address specifications and data types for integer file addresses.

Table 3–32 Integer File Addresses

Address	Format	Comment
mNf:nn	Word	Integer file word
mNf:nn	Longword[1]	Integer file longword
mNf:nn/bb	Bit	Integer file bit

[1]Two words are read as one longword. The low order word is the least significant word.

Floating Point File Addresses

Table 3–33 describes address specifications and data types for floating point file addresses. Floating point addresses are supported only for the SLC-5/03 and SLC-5/04.

Table 3–33 Floating Point File Addresses

Address	Format	Comment
mFf:nn	Float	Float file floating point

ASCII File Addresses

Table 3–34 describes address specifications and data types for ASCII file addresses. ASCII file addresses are supported only for the SLC-5/03 and SLC-5 /04. ASCII addresses must be read from and written to with element counts that are multiples of 2.

Table 3–34 ASCII File Addresses

Address	Format	Comment
Af:nn	String:2	2 ASCII file characters

String Data Addresses

Table 3–35 describes address specifications and data types for string data addresses. String file addresses are supported only for the SLC-5/03 and SLC-5 /04. String file addresses must be read and written to with element counts that are multiples of 2.

Table 3–35 String Data Addresses

Address	Format	Comment
mSTf:nn	String	Entire string structure

3.2.5 Read Status

The read status function issues a diagnostic status request to the PLC, interprets the device response, and returns the interpretation as a character buffer. An error message is displayed if a device definition does not match the device in the device response. If this occurs, you must correct the device definition before the device can be uploaded or downloaded.

Example 3–5 shows an example of the values returned in the status buffer when you enter the READ STATUS command for PLC-5 family devices. The Example 3–6 shows an example of the values returned in the status buffer when you enter the READ STATUS command for SLC family devices. *BASEstar Classic Command Line Interface User's Guide* gives detailed information about the READ STATUS command. The information on the screen varies for each type of device.

Values for the devices calling the ILAN$READ_DEVICE_STATUS routine using the /QUALIFIER=binary qualifier are returned in the status buffer. Refer to the *Allen-Bradley Data Highway/Data Highway Plus Protocol and Command Set* for these values. The SLC family of controllers does not support the /QUALIFIER=binary qualifier.

To display the returned values for device status, enter the following at the BASEstar Classic device connection management prompt:

```
DCM> READ STATUS device-name/FULL
```

Example 3–5 Read Status Screen

```
Status of Device AB_5 at dd-mmm-yyy hh:mm:ss.ss

Interface:   PLC-5 processor interface
Processor:   PLC-5 processor interface
Series/Revision:  B/E
Keyswitch:   Remote program load
Memory Size:   13824
Data Files:   14
Program Files:   4
Processor Number: 1
I/O Address:   0
Memory Protect:   Disabled
Forcing Status:   No forces present
```

Example 3–6 Read Status Screen for SLC

```
Status of device  SLC_504  at dd-mmm-yyyy hh:mm:ss.ss

Interface:                DH+ via Domino Plug
Processor:                SLC-5/04S processor
Series/Revision:          B/3
Processor error word:     0
Processor program ID:     53606
Processor size:           16 K
Processor mode:           Program
Communication status:     Active
```

3.3 Automatic Data Collection

The following sections discuss unsolicited data collection and pollsets for the Allen-Bradley PLCs.

3.3.1 Unsolicited Data Collection

BASEstar Classic device connection management has the ability to receive data values generated by plant devices without issuing a prior 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 PLCs are capable of sending unsolicited messages to an OpenVMS host. In order to generate unsolicited data, the ladder logic program for the Allen-Bradley PLCs must use the Allen-Bradley message instruction. The following components of the message instruction support unsolicited data:

- Address and size of the data to be sent
- Data Highway address of the communications controller to which the OpenVMS host is connected
- Destination (of the data to be sent) table address

The messages sent must be a PLC-2 protected or unprotected write instruction. The address is an octal PLC-2 address.

For SLC controllers use the 485CIF target device to send PLC-2 messages. Also for SLC controllers the destination address is specified as a decimal byte offset. This value must be converted to a word octal address to obtain the unsolicited ID to use for the message coming from the device.

The DAS for AB Data Highway Protocol uses the destination table address as the unsolicited ID. Because unsolicited ID's are numeric, the family of the destination device must be PLC-2. For further information about the message instruction, see the Allen-Bradley documentation of your specific device.

The DAS for AB Data Highway Protocol allows you to define BASEstar Classic device connection management physical points for collecting unsolicited data. When the device sends an unsolicited message to the OpenVMS host, the DAS for AB Data Highway Protocol forwards the message to BASEstar Classic device connection management. The maximum unsolicited message size allowed is 244 bytes. By default, the maximum unsolicited ID designation allowed is 127 (decimal). You can override this default by setting one of the following BASEstar Classic parameters.

PE	Parameter Name
AB	ILAN$AB_MAX_UNSOL_ID
AB2	ILAN$AB2_MAX_UNSOL_ID
AB3	ILAN$AB3_MAX_UNSOL_ID
AB5	ILAN$AB5_MAX_UNSOL_ID
SLC	ILAN$SLC_MAX_UNSOL_ID

The parameter must be within the range 1 - 65535.

Unsolicited messages are received in the DAS for AB Data Highway Protocol by posting an unsolicited read request for the device. When the device sends an unsolicited message to the DAS for AB Data Highway Protocol, the message is read and sent to BASEstar Classic device connection management and any interested applications. It is possible for the device to send another message before the first one has been processed. Only five unsolicited read requests are queued per device. If six messages are read in quick succession before any of the first five have been processed, the sixth message is discarded. The number of queued unsolicited read requests can be redefined by setting one of the following BASEstar Classic parameters:

PE	Parameter Name
AB	ILAN$AB_UNSOL_QUEUE_SIZE
AB2	ILAN$AB2_UNSOL_QUEUE_SIZE
AB3	ILAN$AB3_UNSOL_QUEUE_SIZE
AB5	ILAN$AB5_UNSOL_QUEUE_SIZE
SLC	ILAN$SLC_UNSOL_QUEUE_SIZE

There is no minimum or maximum number of read requests for the queue size.

Note

Increasing the queue size increases the amount of SPT blocks used by the PE. See Section 2.4.2 to see how to calculate the number of SPT blocks used. The increase does not take effect until the device is MODIFIED /DISABLED and MODIFIED/ENABLED using BASEstar Classic device connection management.

For information on setting parameter values, refer to the *BASEstar Classic Configuration and Tuning Guide*.

3.3.2 Pollsets

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

- Physical points with different 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.
- Members of pollsets created for the PLC-3 must have identical context, section, and file values.
- Pollset members for PLC-5s and SLCs must have identical file numbers. PLC-5/250 pollset members must have identical module numbers.

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

A
Logged Messages

The messages in the following sections are logged to the BASEstar Classic history file by the DAS for AB Data Highway Protocol. 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 Data Highway Protocol use event class 21 and event type 25.

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

```
$ BSTAR
BSTAR> SHOW HISTORY/EVENT=21.25.*
```

A.1 NI Logged Messages

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

Hangup occurred on the line.

Explanation: Informational. If the port is a LAT port, then either the network connection to the terminal server has been broken or the serial line connection to the PLC has been disconnected. If the port is a TT port, then the serial connection to the PLC has been disconnected.

User Action: If the cause of the hangup was a network disconnection, then determine the cause of the network disconnection (could be physical problems with the network or the terminal server was powered down) and correct the problem. If the cause of the hangup was the port, then check for broken or missing cables and correct any problems found.

Station is being removed.

Explanation: Informational. This message logs the identity of a station being disconnected due to a hangup on the line.

User Action: None.

Did not find DLE ETX within buffer limit.

Explanation: Error. A message was received on the data highway, but no DLE ETX pair was found before the end of the buffer. Normally, a data highway message is framed by a DLE STX msg DLE ETX pair. In this case the DLE ETX is missing. The cause for this occurrance could be dropped or garbled characters on the line. If the problem persists, there is likely a problem with the cable that connects the data highway to the host.

User Action: Check for frayed wires, improper shielding, improper cable length or routing of the cable near high noise generating equipment such as welders. If using a terminal server, check the terminal server counters for errors such as overruns or parity errors.

Unexpected ETX received. Message discarded.

Explanation: Error. An ETX was received when expecting an ACK, NAK or ENQ character. An ETX should not be received at this point, so the message is discarded. The message that was received is also logged to the history file. The likely cause for this error is either a noisy cable that connects the data highway to the host or, if using a terminal server, an overloaded or faulty network that is causing unusual delays in transmitting and receiving data.

User Action: Check for frayed wires, improper shielding, improper cable length or routing of the cable near high noise generating equipment such as welders. If using a terminal server, check the terminal server counters for errors such as overruns or parity errors.

Possible DLE ACK or DLE NAK embedded in packet - !UB.

Explanation: Error. There is either a DLE ACK or DLE NAK sequence embedded in the packet received from the device. (The actual characrter (ACK or NAK) is logged.) Since the DAS does not support embedded responses other errors are likely to be generated.

User Action: Check the communications card being used to communicate to the data highway and ensure that the embedded responses switch is set to OFF.

Unmatched trans number received for station, tns # = !UW.

Explanation: Error. A response message was received from a PLC and the transaction number in the message does not match any current outstanding transactions. The transaction may have already timed out or this transaction may be a duplicate of one already received. If the transaction has already timed out, then messages are being received too late due either to data highway or, if using a terminal server, to ethernet network delays.

User Action: First determine the cause of the error. If the error is due to a transaction timing out, then the timed out transaction should also have been logged. In this case, determine the cause of the delays in the network, and correct the problem.

Discarded unsolicited message. Unsolicited read queue exhausted.

Explanation: Error. The DAS maintains a queue of buffers to receive unsolicited messages. The size of this queue defaults to a value of "5", but can be increased (or decreased) by creating and setting the parameter ILAN$ABX_UNSOL_QUEUE_SIZE. The name of the parameter is different for each PLC family. This error indicates that all buffers in the queue have been used and no more are available, so the incoming transaction has been discarded and is lost. The cause of the error is either too small a queue size or messages are being received faster than they can be processed.

User Action: Either increase the size of the queue, decrease the number of messages being transmitted, or increase the processing speed of the host.

Discarded unsolicited message. Station not configured.

Explanation: Error. An unsolicited message was received from a PLC station that has not been configured in BASEstar software. This message can also occur for a station that has been configured during startup of the system if a message is received after the line has been configured but before all the devices on the line have reached the "connected" state.

User Action: Configure a device for the station sending the messages. If the message occurs during startup only, then no action needs to be taken.

Data highway error - status code = x!XB.

Explanation: Error. An error has been returned by data highway. The status code returned is logged in the message.

User Action: Look up the error in *Allen-Bradley Data Highway/Data Highway Plus Protocol and Command Set* and take action based on the error found. Also, examine the log file for any extended status codes that may be logged as well.

Data highway error - extended status code = x!XB.

Explanation: Error. An error has been returned by data highway. The extended status code returned is logged in the message.

User Action: Look up the error in *Allen-Bradley Data Highway/Data Highway Plus Protocol and Command Set* and take action based on the error found.

NAK sent. BCC did not match, expected x!XB, received x!XB.

Explanation: Error. The checksum calculated by the DAS and the checksum received in the message did not match. The DAS responded to the message by sending a "NAK". This message is probably caused by a noisy line that is either dropping or inserting characters into the message.

User Action: Check for frayed wires, improper shielding, improper cable length or routing of the cable near high noise generating equipment such as welders. If using a terminal server, check the terminal server counters for errors such as overruns or parity errors.

Bad QIO read status,

Explanation: Error. An error occurred when reading a message from the line. The status code returned from the OpenVMS system service QIO call is logged along with this message.

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

Bad QIO write status,

Explanation: Error. An error occurred when writing a message to the line. The status code returned from the OpenVMS system service QIO call is logged along with this message.

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

Unexpected ACK received. No request pending.

Explanation: Error. An ACK was received when there is no write currently pending. The most likely cause for this error is that the write has already timed out. The timeout can be caused by a busy data highway or a loaded network, if connected via a terminal server.

User Action: The default timeout for the DAS is 3 seconds. The timeout value can be increased by modifying the /TIMEOUT parameter on the path definition. Alternatively, the response time can be shortened by decreasing the load on either the data highway network or the ethernet network, if connected via a terminal server.

Timed out waiting for transaction = !UW.

Explanation: Error. A pending transaction timed out. The timeout time is 3 times the read timeout set in the /TIMEOUT parameter on the path definition. (If the /TIMEOUT parameter is not set, then the timeout defaults to 3 seconds, and the transaction timeout is 9 seconds.) If the transaction is arriving late another message is logged stating that no matching transaction number is found. In this case, it is likely that either the communication card is having trouble processing all the transactions on the data highway or there are significant errors or retries that are slowing down the effective transaction rate.

User Action: The transaction timeout can be indirectly modified by increasing the /TIMEOUT parameter on the path definition. If the transaction is arriving late, then determine where the bottleneck in communications is occurring (either the data highway communications card, the ethernet network or the host processor) and take actions to reduce the load.

Unexpected NAK received. No request pending.

Explanation: Error. A NAK was received when there is no write currently pending. The most likely cause for this error is that the write has already timed out. The timeout can be caused by a busy data highway or a loaded network, if connected via a terminal server.

User Action: The default timeout for the DAS is 3 seconds. The timeout value can be increased by modifying the /TIMEOUT parameter on the path definition. Alternatively, the response time can be shortened by decreasing the load on either the data highway network or the ethernet network, if connected via a terminal server.

Discarded garbage data on line - message lost.

Explanation: Error. An unexpected protocol control sequence was received, so the message is discarded. The message that was received is also logged to the history file. The likely cause for this error is either a noisy cable that connects the data highway to the host or, if using a terminal server, an overloaded or faulty network that is causing unusual delays in transmitting and receiving data.

User Action: Check for frayed wires, improper shielding, improper cable length or routing of the cable near high noise generating equipment such as welders. If using a terminal server, check the terminal server counters for errors such as overruns or parity errors.

Discarded unsolicited message. Unsupported command = x!XB sent by PLC.

Explanation: Error. An unsolicited message was received from a PLC, but the command in the message is one that is not supported by the DAS. The command code that is being sent by the PLC is logged in the message. The DAS only supports command codes for PLC-2 protected and unprotected writes.

User Action: Reprogram the PLC to send one of the two command codes supported by the DAS.

Discarded unsolicited message. Unsupported function = x!XB sent by PLC.

Explanation: Error. An unsolicited message was received from a PLC, but the function in the message is one that is not supported by the DAS. The function code that is being sent by the PLC is logged in the message. The DAS only supports command codes for PLC-2 protected and unprotected writes.

User Action: Reprogram the PLC to send one of the two command codes supported by the DAS.

No license is active for this software product.

Explanation: Error. The DAS could not find a license for the product.

User Action: Register and load a valid license for this DAS.

A.2 PE Logged Messages

The following messages are logged by one or more of the Protocol Emulators (PEs).

Unsolicited message received. ID !OW out of range.
Current limit is !OW.

Explanation: Error. An unsolicited message was received from a PLC, but the unsolicited ID (PLC-2 address) was too large for the DAS. By default, the largest value for the unsolicited ID is 127 (decimal). This value can be increased by creating and setting the parameter ILAN$ABX_MAX_UNSOL_ ID. The name of the parameter is different for each PLC family.

User Action: Create and set the ILAN$ABX_MAX_UNSOL_ID parameter to be larger than the unsolicited ID received or change the PLC logic to use a different PLC-2 address. The device must be disabled and then reenabled to use the parameter.

Unsolicited message received. No matching phypoint found for ID !OW.

Explanation: Error. An unsolicited message was received from the PLC, but a physical point with a matching unsolicited ID was not found for this device.

User Action: Create a physical point with a matching ID and format for the data received from the PLC, or change the PLC logic to send an unsolicited message with the unsolicited ID and format of an existing physical point.

Phypoint unsolicited ID !OW - out of range.
Current limit is !OW.

Explanation: Error. A phypoint has been created, but the unsolicited ID (PLC-2 address) was too large for the DAS. By default, the largest value for the unsolicited ID is 127 (decimal). This value can be increased by creating and setting the parameter ILAN$ABX_MAX_UNSOL_ID. The name of the parameter is different for each PLC family. If the device is disabled when the phypoint is created, this message is logged when the device is enabled.

User Action: Create and set the ILAN$ABX_MAX_UNSOL_ID parameter to be larger than the unsolicited ID used or change the physical point to use a different PLC-2 address for the unsolicited ID. The device must be disabled and then reenabled to use the new parameter.

Phypoint unsolicited ID !OW - already defined.

Explanation: Error. A phypoint has been created, but the unsolicited ID (PLC-2 address) is already in use by another phypoint. Unsolicited IDs must be unique for this DAS.

User Action: Create the phypoint using a different unsolicited ID (PLC-2 address).

Attempt to download !AD file to !AD.

Explanation: Error. An attempt was made to download a file that was uploaded from a PLC with a different processor type. Only files with the same processor type as the PLC being downloaded to can be downloaded.

User Action: Download a file with a processor type that matches the type of the PLC being downloaded to.

PLC memory too small to fit the requested file.
PLC logic file memory = !UL, PLC memory = !UL.

Explanation: Error. The size of memory in the file being downloaded is greater than the size of memory in the PLC being downloaded to.

User Action: Download a file with a memory image that is less than or equal to the actual size of the memory in the PLC.

Requested download file is not a valid PLC logic file.

Explanation: Error. The file was missing header information that identified the file as a valid PLC logic file.

User Action: Download a file that is valid to be downloaded to this PLC.

Data table size mismatch, file = !UW, PLC = !UW.

Explanation: Error. When attempting to download to a PLC-2 where the interface module is a 1771-KA revision A or 1771-KG revision A, the data table sizes did not equal each other.

User Action: Download a compatible PLC-2 file to the PLC-2.

Requested download !AD file, processor type mismatch.

Explanation: Error. An attempt was made to download a file that was uploaded from a PLC-5 with a different processor type and the OVERRIDE qualifier was not specified.

User Action: Download a file with a processor type that matches the type of the PLC being downloaded to or use the OVERRIDE qualifer to download a compatible PLC logic file.

Downloading !AD file to !AD.

Explanation: Informational. A file with a different processor type is being downloaded to a PLC-5 processor.

User Action: None.

Index