

Wonderware®

ABDHPLUS DAServer User's Guide

Version 1.5

Last Revision: 2/5/07

Invensys Systems, Inc.

All rights reserved. No part of this documentation shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Invensys Systems, Inc. No copyright or patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this documentation, the publisher and the author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

The information in this documentation is subject to change without notice and does not represent a commitment on the part of Invensys Systems, Inc. The software described in this documentation is furnished under a license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of these agreements.

© 2002, 2007 Invensys Systems, Inc. All Rights Reserved.

Invensys Systems, Inc.
26561 Rancho Parkway South
Lake Forest, CA 92630 U.S.A.
(949) 727-3200

<http://www.wonderware.com>

For comments or suggestions about the product documentation, send an e-mail message to productdocs@wonderware.com.

Trademarks

All terms mentioned in this documentation that are known to be trademarks or service marks have been appropriately capitalized. Invensys Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this documentation should not be regarded as affecting the validity of any trademark or service mark.

Alarm Logger, ActiveFactory, ArchestrA, Avantis, DBDump, DBLoad, DT Analyst, FactoryFocus, FactoryOffice, FactorySuite, FactorySuite A², InBatch, InControl, IndustrialRAD, IndustrialSQL Server, InTouch, MaintenanceSuite, MuniSuite, QI Analyst, SCADAAlarm, SCADASuite, SuiteLink, SuiteVoyager, WindowMaker, WindowViewer, Wonderware, and Wonderware Logger are trademarks of Invensys plc, its subsidiaries and affiliates. All other brands may be trademarks of their respective owners.

Contents

Before You Begin 7

About This Book 7

CHAPTER 1: Introduction.....9

Overview 9

Communications Protocols 10

 Application Communications Protocols 10

 Bus Communications Protocols..... 11

Accessing Items via the DAServer..... 12

Features 13

Demo Mode..... 13

CHAPTER 2: Configuration 15

Installing the Card Driver..... 15

Getting Started Quickly with the DAServer 17

Configuring the DAServer 20

ABDHPLUS Hierarchy in the DAServer Manager 22

 CARD Objects 22

 PKT_CARD Object 23

 SSTPCI_CARD Object 25

 PLC and Bridge-Module Objects 26

 PLC5_DHP Object 27

 SLC500_DHP Object 28

 M1785KEBR Object 29

 M1785KA Object 29

 M1785KA5 Object 31

Configuring Device Group Definitions..... 32

 Device Group Definitions..... 32

 Scan-Based Message Handling 33

 Unsolicited Message Handling..... 34

Archiving Configuration Sets..... 35

Hot Configuration 36

CHAPTER 3: Item Names.....37

PLC-5 Item Naming..... 37

 Output File Items 38

 Input File Items..... 39

 Status File Items 39

 Binary File Items 40

 Timer File Items 40

Counter File Items	41
Control File Items.....	41
Integer File Items.....	42
Floating Point File Items	42
ASCII File Items	42
BCD File Items.....	43
ASCII String Section Items	43
Block Transfer Section Items	44
PID Section Items.....	44
SFC Status Section Items	45
SLC 500 Item Naming	45
Output File Items.....	46
Input File Items	46
Addressing SLC I/O Modules	47
Status File Items	49
Binary File Items.....	50
Timer File Items	50
Counter File Items.....	51
Control File Items.....	51
Integer File Items.....	52
Floating Point File Items	52
ASCII File Items	52
ASCII String Section Items.....	53
DAServer Standard System Items.....	53
DAServer Global System Item.....	54
DAServer Device-Specific System Items	54
DAServer Device-Group-Specific System Items.....	56
Generic OPC Syntax	58

CHAPTER 4: Troubleshooting61

Monitoring Connectivity Status with the PLC.....	61
Monitoring the Status of DAS Conversations.....	62
Using DDEStatus and IOStatus in Excel.....	62
Reading Values from the DAServer into Excel.....	63
Writing Values to the DAServer from Excel	63
Error Messages, Codes, and Conditions	64
DAServer Error Messages.....	64
PLC-5 Error Messages	79
SLC 500 Error Messages.....	80
PLC-5 and SLC 500 Error Messages	81
ABDHPLUS DAServer Error Codes	82
Data Highway Plus Error Conditions.....	83

CHAPTER 5: Reference85

DAServer Architecture	85
DAServers	85

Plug-ins 86
DAS Engine 87
PLC Protocol Layer 87
Component Environments..... 87

Index 89

Before You Begin

About This Book

This user's guide describes the user interface and functions of the Wonderware® ABDHPLUS DAServer™. It also provides you with the step-by-step procedures on how to configure and use the ABDHPLUS DAServer after it is installed.

The ABDHPLUS user's guide is organized in the following fashion:

- **Contents**
- **Introduction:** contains overview information about the ABDHPLUS DAServer and the environment in which it works.
- **Configuration:** contains a detailed description of the user interface elements of this DAServer in addition to its functionality.
- **Item Names:** describes the item naming conventions for targeted devices.
- **Troubleshooting:** provides information about error messages, codes, and conditions displayed by the ABDHPLUS DAServer.
- **Reference:** describes DAServer architecture in general.
- **Index**

You can view this document on-line or you can print it, in part or whole, by using the Adobe Acrobat Reader's print facility. To view this document properly, you must use version 4.0 of the Acrobat Reader.

CHAPTER 1

Introduction

This chapter provides an overview of the Wonderware ABDHPLUS DAServer, including application and bus-level communications protocols, item naming conventions, and the DAServer features.

Contents

- Overview
- Communications Protocols
- Accessing Items via the DAServer
- Features
- Demo Mode

Overview

The Wonderware® ABDHPLUS DAServer, as part of the Rockwell_DASpak, is the next-generation server for the current Wonderware Allen-Bradley 1784-KT and SST 5136-SD I/O Servers.

This DAServer is a Microsoft® Windows® application program designed to provide direct and indirect connectivity to the Allen-Bradley families of PLC-5 and SLC 500 controllers. The DAServer acts as a communications protocol server and allows other Windows application programs access to data in the Allen-Bradley PLC-5 and SLC 500 PLCs (also referred to as devices), which are connected to the Data Highway Plus Bridge Module (DH485, DH, or DF1). The DAServer does not require the Rockwell Software RSLinx™ OEM package.

While the DAServer is primarily intended for use with Wonderware InTouch® (version 7.11 Patch 02 and later), it may be used by any Microsoft Windows program capable of acting as a DDE, FastDDE, SuiteLink™, or OPC client.

Communications Protocols

The ABDHPLUS DAServer (Data Access Server) communicates with clients and PLCs using different communications protocols. The DAServer uses application protocols (such as OPC, DDE, and SuiteLink) to communicate with the clients, and Data Highway Plus direct connection or with the Bridge Modules (DH485, DH, or DF1) to communicate with the Allen-Bradley PLC-5 and SLC 500 controllers.

For more information about the DAServer architecture, see the Reference section.

Application Communications Protocols

OPC (OLE for Process Control) is a non-proprietary set of standard interfaces based upon Microsoft's OLE/COM technology. This standard makes possible interoperability between automation/control applications, field systems/devices and business/office applications. Avoiding the traditional requirement of software/application developers to write custom drivers to exchange data with field devices, OPC defines a common, high-performance interface that permits this work to be done once, and then easily reused by HMI, SCADA, control and custom applications. Over the network, OPC uses DCOM (Distributed COM) for remote communications.

SuiteLink uses a TCP/IP-based protocol and is designed specifically to meet industrial needs such as data integrity, high throughput, and easier diagnostics. This TCP/IP standard is supported on Windows NT and Windows NT-technology-based operating systems (for example, Windows NT, Windows 2000, and Windows XP).

SuiteLink is not a replacement for DDE, FastDDE, or NetDDE. The protocol used between a client and a server depends on your network connections and configurations. SuiteLink provides the following features:

- Value Time Quality (VTQ) places a time stamp and quality indicator on all data values delivered to VTQ-aware clients.
- Extensive diagnostics of the data throughput, server loading, computer resource consumption, and network transport are made accessible through the operating system's performance monitor. This feature is critical for the operation and maintenance of distributed industrial networks.
- Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large node count.
- The network transport protocol is TCP/IP using Microsoft's standard WinSock interface.

FastDDE provides a means of packing many proprietary Wonderware Dynamic Data Exchange messages into a single Microsoft DDE message. This packing improves efficiency and performance by reducing the total number of DDE transactions required between a client and a server. Although Wonderware's FastDDE has extended the usefulness of DDE for our industry, this extension is being pushed to its performance constraints in distributed environments.

DDE is a communications protocol developed by Microsoft to allow applications in the Windows environment to send/receive data and instructions to/from each other. It implements a Client/Server relationship between two concurrently running applications. The server application provides the data and accepts requests from any other application interested in its data. Requesting applications are called clients. Some applications such as InTouch and Microsoft Excel can simultaneously be both a client and a server.

NetDDE is a communications protocol that extends the standard DDE functionality to include communications over local area networks and through serial ports. Network extensions are available to allow DDE links between applications running on different computers connected via networks or modems. For example, NetDDE supports DDE between applications running on IBM-compatible computers connected via LAN or modem, and DDE-aware applications running on non-IBM-compatible computers under operating environments such as VMS and UNIX.

Bus Communications Protocols

Several types of bus communications protocols are utilized by the ABDHPLUS DAServer to communicate with the controllers.

- Data Highway Plus
- Bridge Module
 - DH-485
 - DH
 - DF1

Note DH485, DH, and DF1 are only supported in the bridge modules.

The Data Highway Plus is a Local Area Network (LAN) that allows peer-to-peer communications among a maximum of 64 nodes. It is used when a connection of a small number of nodes on a common link or one network is needed. The Data Highway Plus is used by both the PLC-5 and SLC 500 families of controllers for a connection to the adapter card.

The other type of module the controllers are connected to is the Bridge Module. The bridge module provides communications to the Allen-Bradley PLC-5 or SLC 500 families of controllers that are located in two different networks, connected to one or another to the PCI adapters (1784-PKTx or 1784-PKTxD or SST 5136-SD), through various Data Highway Plus, DH485, DH, RS-232 asynchronous link, using Allen-Bradley bridge modules (1785-KA5, 1785-KA, or 1785-KE).

Important! The DAServer will support PCI adapters in any particular system. The DAServer will **not** support a mixture of cards, from different vendors, in the same computer.

Accessing Items via the DAServer

The method for accessing items through the DAServer depends on the communications protocol being used.

In the case of OPC communications, the protocol addresses an element of data in a conversation with six characteristics: node name, program name, group name, device group, link name, and item name.

- The node name (required for remote access) and device group are optional.
- A fully qualified OPC Item name (ItemID) is composed of the link name and item name.
- All other characteristics are specified through separate DAServer means.

To access an OPC item, the OPC client needs to connect to the DAServer (either in-process or out-of-process) and create an OPC group defining the data-acquisition properties for the collection of items to be added. OPC groups can be either public or private. Public OPC groups are shared across multiple clients, whereas private OPC groups are local to a single client. Optionally a device group, which indicates the access path to the items for read/write, can be specified from the DAServer.

The following briefly describes each characteristic of the OPC protocol:

- **node name:** Computer (host) name identifying a specific node on the network (for Remote Access ONLY).
- **program name:** The registered OPC server name uniquely identifying a specific server (ProgID). For this DAServer, the program name is **ArchestrA.DASABDHPLUS.1**.
- **group name:** The OPC group created from the client for organizing a collection of items logically with the same data-acquisition properties between the client and the server, such as update rate.
- **device group:** Meaningful names configured in the DAServer under a specific controller for the common custom attributes between the DAServer and the device, such as update interval. If not specified from the client, the default device group using the global-configuration attribute values from the DAServer is assumed. Functionally, a device group is equivalent to an access path (optional).
- **link name:** The set of hierarchy node names, representing the specific devices on a communications path link from the hierarchy root to a specific controller as configured for this DAServer under the DAServer Manager, separated by delimiters.
- **item name:** A specific data element, the leaf of the hierarchy tree of this DAServer, within the specified group. For example, when using this DAServer, an item can be a relay, timer, counter, register, and so on., in the controller.

In the case of DDE/SuiteLink communications, the protocol addresses an element of data in a conversation that uses a four-part naming convention that includes the node name, application name, topic name, and item name. The fully qualified DDE/SuiteLink naming convention includes all four parts, although the node-name part (required for remote access only) is optional.

The following briefly describes each portion of this naming convention:

- **node name:** Computer (host) name identifying a specific node on the network (for Remote Access ONLY).
- **application name:** The name of the Windows program (this DAServer) that will be accessing the data element. In the case of data coming from or going to Allen-Bradley devices via the DDE/SuiteLink PlugIn of this DAServer, the application name portion of the address is **DASABDHPLUS**.
- **topic name:** Meaningful names are configured in the DAServer to identify specific devices. These names are then used as the topic names in all conversations with that device. For example, **ABPLC**. Topic name maps to a device group defined in the DAServer.

Note You can define multiple device group (topic) names for the same device (PLC) to poll different points at different rates.

- **item name:** A specific data element within the specified topic. For example, when using this DAServer, an item can be a relay, timer, counter, register, and so on, in the PLC.

Note The term "point" is used interchangeably with the term "item" in this user's guide.

For more information on item/point names, see the "Item Names" section of this user's guide.

Features

The Wonderware ABDHPLUS DAServer provides the following features:

- The ability to communicate over multiple application-level protocols at the same time.
- The ability to add new application-level protocols on the fly.
- The ability to be configured remotely.
- New, robust diagnostic abilities.

For more in-depth information on the DAServer architecture, see the Reference section.

Demo Mode

You can install a fully functioning version of this DAServer for demonstration purposes without a license. Demo mode allows you to test the functionality of the DAServer for 120 minutes. After that time, you must install a license to continue using the DAServer.

When you first start this DAServer, it checks for a license. If the DAServer cannot find a valid license installed on the local computer, it logs a warning message indicating a valid license cannot be retrieved, and enters Demo mode. Thereafter, the DAServer repeats its request for the license every 30 seconds. If no license is found, the DAServer again logs a warning message on the issue.

This process is repeated for 120 minutes, after which the DAServer stops updating Read/write on all device items (read from cache is allowed, but all non-system data would receive Bad quality status). The DAServer continues to request for a license. Clients continue to function normally (for instance, you can still add or remove an item, but its quality is set to Bad until a license is obtained).

Note Use the `$$SYS$Licensed` system item, a read-only Boolean item, to check the status of your license: True for Licensed and False for Not Licensed.

If you subsequently add a license to the License Manager, the DAServer logs a message acknowledging the license, switches out of Demo mode, and runs normally.

Note Once a DAServer obtains a valid license, it no longer checks for a license. Thus, if your license expires, your DAServer would cease to function but this condition would not be logged until the next restart of the DAServer.

CHAPTER 2

Configuration

After the adapter-card installation, the driver for the card needs to be installed. In this section, a step-by-step procedure for this card-driver installation is included.

Once the Wonderware ABDHPLUS DAServer has been installed, a small amount of configuration is required. This configuration is performed using the **DAServer Manager** hosted in the System Management Console after it is started through the **Programs** menu of the Windows **Start** button.

Before the DAServer is activated, the device hierarchy, simulating the physical hardware layout, must first be built to establish communications to each of the controllers. Once the ABDHPLUS hierarchy has been built, the respective devices for communications can be configured. Finally, the desired Device Groups for each controller may be created.

Contents

- Installing the Card Driver
- Getting Started Quickly with the DAServer
- Configuring the DAServer
- ABDHPLUS Hierarchy in the DAServer Manager
- Configuring Device Group Definitions
- Archiving Configuration Sets
- Hot Configuration

Installing the Card Driver

After the adapter card (either PKTX/PKTXD card or SSTPCI card) has been installed, the computer will ask for its driver to be installed. The installation of the card driver is performed with the help of the Found New Hardware Wizard.

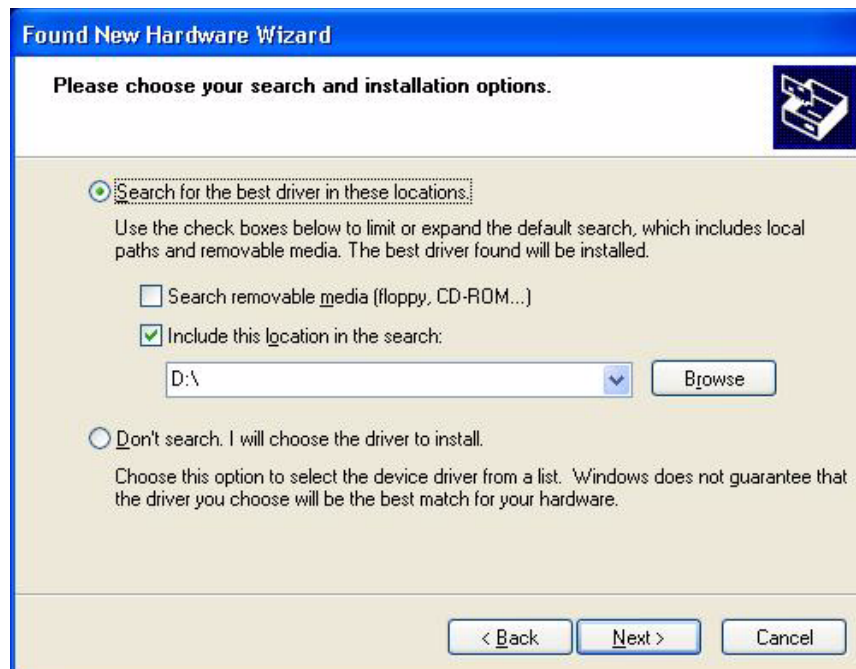
To install the card driver

1. From the Wizard's **Welcome to the Found New Hardware Wizard** page, click on the option button, **Install from a list or specific location (Advanced)**.



2. Click the **Next** button.

- The Wizard will ask you to choose your search and installation options.



3. Enter the path where the driver file is located.

- The following driver files are provided for you.
 - The **ab1784kt.inf** file for the PKT card.

- The **ss5136sd.inf** for the SSTPCI card.
 - These two files are located in the same directory as the **DAServer Setup.exe** file.
4. After the respective driver-file path has been entered, click the **Next** button.
 - The Wizard will let you know that the driver for the card has been installed.



When the card-driver installation is completed, you can start the preparations of the DAServer for use.

Getting Started Quickly with the DAServer

This section briefly describes the procedures required to prepare the ABDHPLUS DAServer for use. Detailed descriptions of each step can be found in the following sections of this documentation. This section is intended for people who are familiar with DAServers.

Note If you are not familiar with the DAServer functionality, please proceed to the more-detailed procedures following this section.

To prepare the ABDHPLUS DAServer

1. Install the Wonderware ABDHPLUS DAServer on Windows by running the **Setup.exe** program.

Note In addition to the DAServer, the drivers of the respective cards are also automatically installed.

Note The DAServer installation instructions are included in a separate Help file (.chm extension).

- Accept all the default settings during installation.
-

Note Since there are no default values for security settings, you must take note of the User Name and password selected during the install.

2. Start the ABDHPLUS DAServer by selecting the **Programs** menu from the **Start** button on the taskbar.
3. Navigate to the **Wonderware** folder that contains the System Management Console, then click **System Management Console**.
4. From the **System Management Console**, find the ABDHPLUS DAServer in the **DAServer Manager** tree, the location in which it is installed.
 - Under the Local Node, the DAServer name is **ArchestrA.DASABDHPlus.1**.
 - See the DAServer Manager Online Help for general information about working in this snap-in environment.
5. The new ABDHPLUS DAServer must now be configured.
 - Before proceeding, determine the hierarchical structure of the network/PLC environment to which you plan to connect.
6. Right-click the **Configuration** branch of the hierarchy, and select one of the two **CARD Objects** (representing the adapter cards for the Allen Bradley or SST processor types):

PKT_CARD object
SSTPCI_CARD object

Note Multiple PKT_CARD or SSTPCI_CARD objects can be created for the DAServer by right-clicking the **Configuration** object and selecting **Add PKT_CARD Object** or **Add SSTPCI_CARD Object** from the shortcut menu.

In this step, in addition to Step 7, the hierarchy entry is added in "edit mode," providing a convenient place for you to appropriately describe components of your specific hardware environment.

- If the object is not renamed at this time, a numeric sequencing system is applied.
 - Any hierarchy entry can be renamed at a later time.
-

Important! The DAServer will support PCI adapters in any particular system. The DAServer will **not** support a mixture of cards, from different vendors, in the same computer.

7. Right-click on the selected **CARD** object, and select the appropriate item from five selections, each of which represents the PLC or bridge module used to connect to the adapter card:

PLC5_DHP object
SLC500_DHP object
M1785KEBR object
M1785KA object
M1785KA5 object

Note Multiple PLC5_DHP, SLC500_DHP, M1785KEBR, M1785KA, and M1785KA5 objects can be created for each CARD object by right-clicking on the selected **New_<Name>_CARD_000** object in the higher tree branch, and selecting **Add <Name> Object** from the shortcut menu, respectively.

- Up to 63 of objects, under the adapter card, can be added to the hierarchy.
- PLC-5 and SLC 500 objects represent the logical end points for the hierarchy being configured; they have no configurable child objects.

Note The Data Highway Plus allows peer-to-peer communications among a maximum of 64 nodes, including the adapter card and bridge module.

8. Configure the respective device objects created in the preceding steps with the appropriate parameter values, if applicable.
9. Optionally, create the desired device groups in the **Device Groups** dialog box of each logical end-point object.

Note When any configuration view is in an open state and you open the same server the second time, the DAServer locks the second instance of this same-server access for any update or configuration activities. Access to this second opening instance will resume after the first one has been closed.

The DAServer is now ready for use. In order to use the DAServer, you must activate it.

- If you are using an OPC Client, the DAServer will auto-start.
- If you are using DDE/SuiteLink, you must start the DAServer either as a manual or automatic service.
- To activate the DAServer, right-click on **Archestra.DASABDHPlus.1** and select **Activate Server** from the shortcut menu.

Note To run the ABDHPLUS DAServer as a service, use the shortcut menu on the **DAServer name** and select **Configure As Service**. You can configure it as an auto service or manual service. For more information about configuring the DAServer as a service, see the Activation/Deactivation/Service Component of the DAServer Manager documentation.

Configuring the DAServer

Note This DAServer is hosted by the DAServer Manager, a Microsoft Management Console (MMC) snap-in, which is part of the ArcestrA System Management Console (SMC) suite of utilities. Many high-level functions and user-interface elements of the DAServer Manager are universal to all DAServers, and **only** the documentation for the DAServer Manager contains descriptions of those universal functions/UI elements. Therefore, reading the documentation for both the MMC and the DAServer Manager is critical to understanding this user's guide. To read the documentation about the MMC and DAServer Manager, right-click on the **DAServer Manager** icon and select the **Help** menu. Both the MMC and DAServer Manager Help is displayed. Select the appropriate Help topics. An Adobe Acrobat version of the DAServer Manager documentation (DAServerManager.pdf) is also provided.

Note The shortcut menu items described in this document typically represent only a subset of any actual shortcut menu. Most items in each shortcut menu are standard Windows commands. For more information about those commands, please see **Help** by right-clicking the **System Management Console** icon.

Before it can be configured, the Wonderware ABDHPLUS DAServer has to be installed on Windows.

To prepare the ABDHPLUS DAServer

1. Install the Wonderware ABDHPLUS DAServer on Windows by running the **Setup.exe** program.

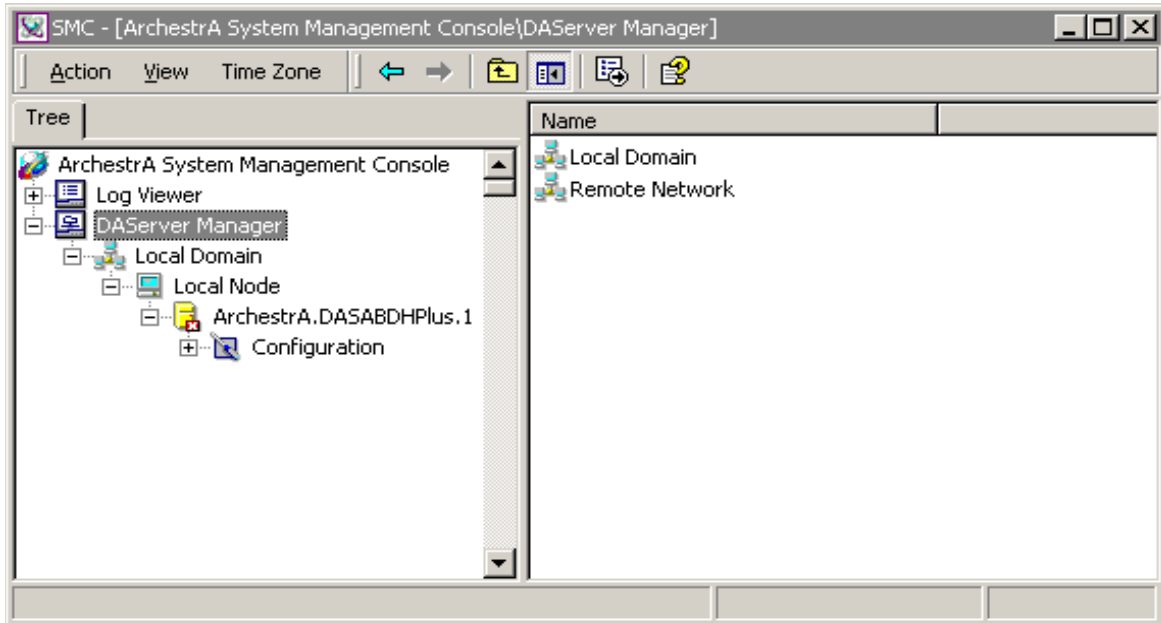
Note In addition to the DAServer, the drivers of the respective cards are also automatically installed.

Note The DAServer installation instructions are included in a separate Help file (.chm extension).

2. Accept all the default settings during installation.

Note Since there are no default values for security settings, you must take note of the User Name and password selected during the install.

3. After it has been installed, start the System Manager Console by clicking the **Start** button on the Windows taskbar, and pointing to **Programs**.
4. Point to the **Wonderware** folder that contains the System Management Console, then click **System Management Console**.
5. From the System Management Console tree, click on **DAServer Manager**.
6. Click on **Local Domain**, then **Local Node**.
 - Under the Local Node, the DAServer name is **ArcestrA.DASABDHPlus.1**.



- See the DA Server Manager documentation for general information about working in this snap-in environment.
7. Before the DA Server is started, the device hierarchy must be built to establish communications to each of the controllers.

Note For step-by-step procedures on building the device hierarchy, please see the section, "ABDHPLUS Hierarchy in the DA Server Manager."

Note Selecting the **Configuration** object of the hierarchy tree displays the global parameters interface for this DA Server. The default Poke Mode settings for the DA Server is Transition mode. Configure all other global parameters as required for this DA Server. For more information about the Global Parameters dialog box, including descriptions of the different Poke Modes, see the DA Server Manager documentation. You can access that documentation by right-clicking on the **DA Server Manager** icon, clicking the Help topics on the **Help** menu, and then navigating through the **DA Server Manager** book.

Important! Any Global Parameters that appear dimmed are not supported.

8. When the ABDHPLUS hierarchy build has been completed, you can start configuring the respective devices for communications.
9. Finally, you may create the desired Device Groups for each controller by:
 - Navigating to the object of interest in the **DA Server Manager** tree view.
 - Clicking on the **Device Groups** tab.

Note For step-by-step procedures on configuring Device Groups, please see the section, "Configuring Device Group Definitions."

Note When any configuration view is in an open state and you open the same server the second time, the DAServer locks the second instance of this same-server access for any update or configuration activities. Access to this second opening instance will resume after the first one has been closed.

The DAServer will be ready to use after it is activated.

- If you are using an OPC Client, the DAServer will auto-start.
- If you are using DDE/SuiteLink, you must start the DAServer either as a manual or automatic service.
- The DAServer can be activated by right-clicking on **ArchestrA.DASABDHPlus.1** and selecting **Activate Server** from the shortcut menu.

Note To run the ABDHPLUS DAServer as a service, right-click on the **DAServer name** and select **Configure As Service** from the shortcut menu. You can configure it as an auto service or manual service. For more information about configuring the DAServer as a service, see the Activation/Deactivation/Service Component of the DAServer Manager documentation.

ABDHPLUS Hierarchy in the DAServer Manager

Note Before attempting to configure your DAServer, you should determine the hierarchical structure of your network/PLC environment.

The server-specific configuration portion of the ABDHPLUS DAServer hierarchy tree under the DAServer Manager starts at the **CARD** object.

- This is the place in the DAServer hierarchy for configuring the adapter cards for the Allen Bradley or SST processor types.
- Rename this object as appropriate.

Important! The DAServer will support PCI adapters in any particular system. The DAServer will **not** support a mixture of cards, from different vendors, in the **same** computer.

Important! If you subsequently clear your configuration hierarchy, you must create each of these **CARD** objects by right-clicking on the **Configuration** object and selecting the **Add <Name>_CARD** Object in the shortcut menu, respectively. A **New_<Name>_CARD_000** object is created. Rename as appropriate. From this point, all of the following instructions apply.

CARD Objects

Two **CARD** objects can be created in the initial level of the ABDHPLUS DAServer hierarchy in the DAServer Manager. Each **CARD** object configuration view is the place in the hierarchy to configure the different adapter cards for the Allen Bradley or SST processor types.

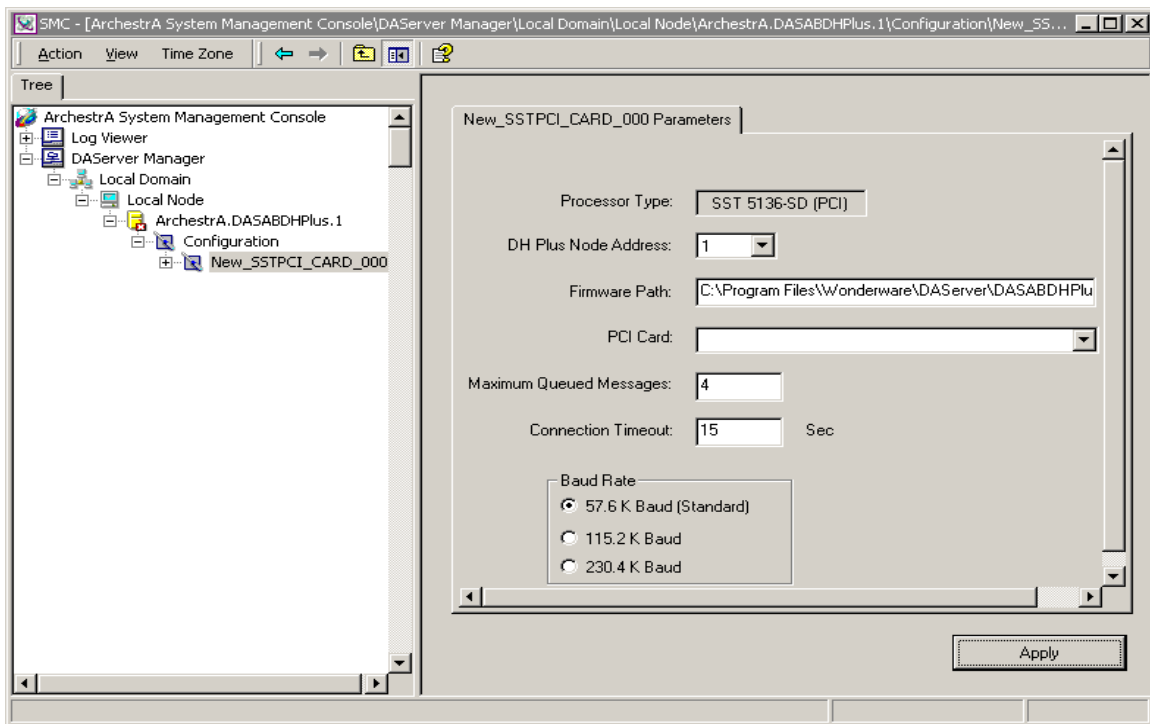
The two CARD objects supported by the DAServer are as follows:

- **PKT_CARD object** – More than one PKT_CARD object can be created, and each one of them represents either the 1784-PKTx or 1784-PKTxD adapter card.
- **SSTPCI_CARD object** – Multiple SSTPCI_CARD objects can be created for the DAServer, with each one of them representing the SST 5136-SD-PCI adapter card.

To create a CARD object from the Configuration branch

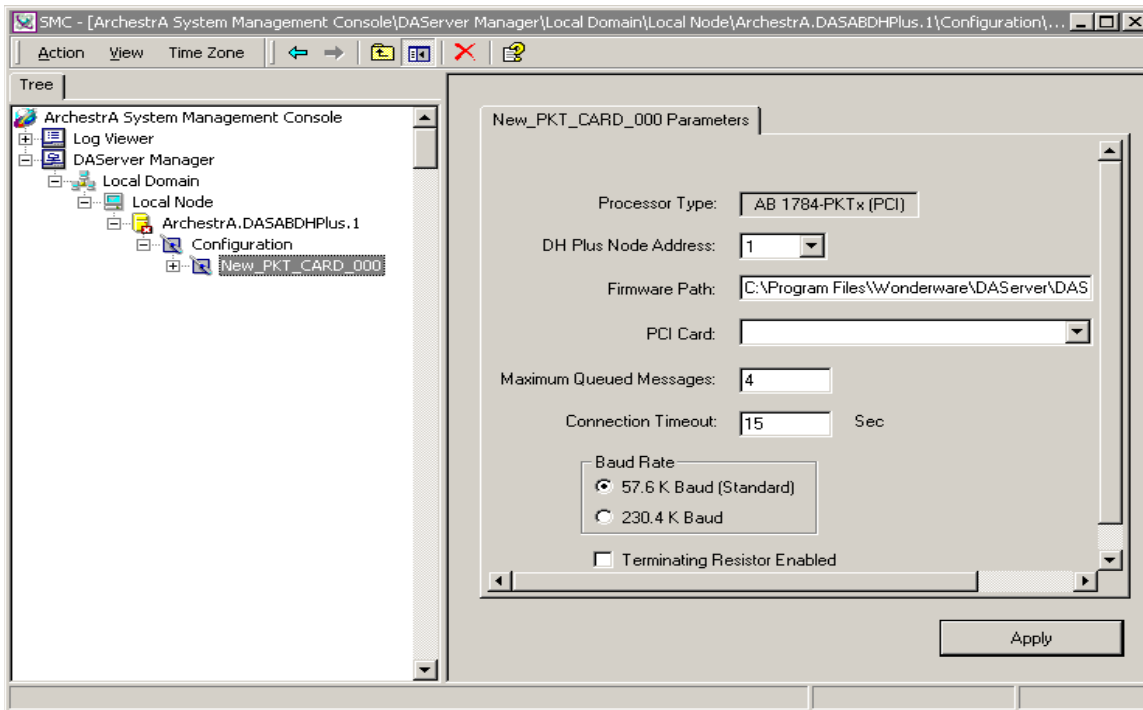
1. Right-click on **Configuration**.
2. Select the appropriate **Add <NAME>_CARD Object** from the shortcut menu.
 - Each object can be renamed when it is created in the hierarchy, or at a later time.
3. Rename each of the newly created object as appropriate.

The following figure shows the CARD objects in the Configuration branch of the hierarchy.



PKT_CARD Object

The **New_PKT_CARD_000 Parameters** configuration view is used to configure the Allen Bradley 1784-PKTx or 1784-PKTxD PCI-type adapters. A maximum of 4 (four) PKT_CARD objects can be configured.



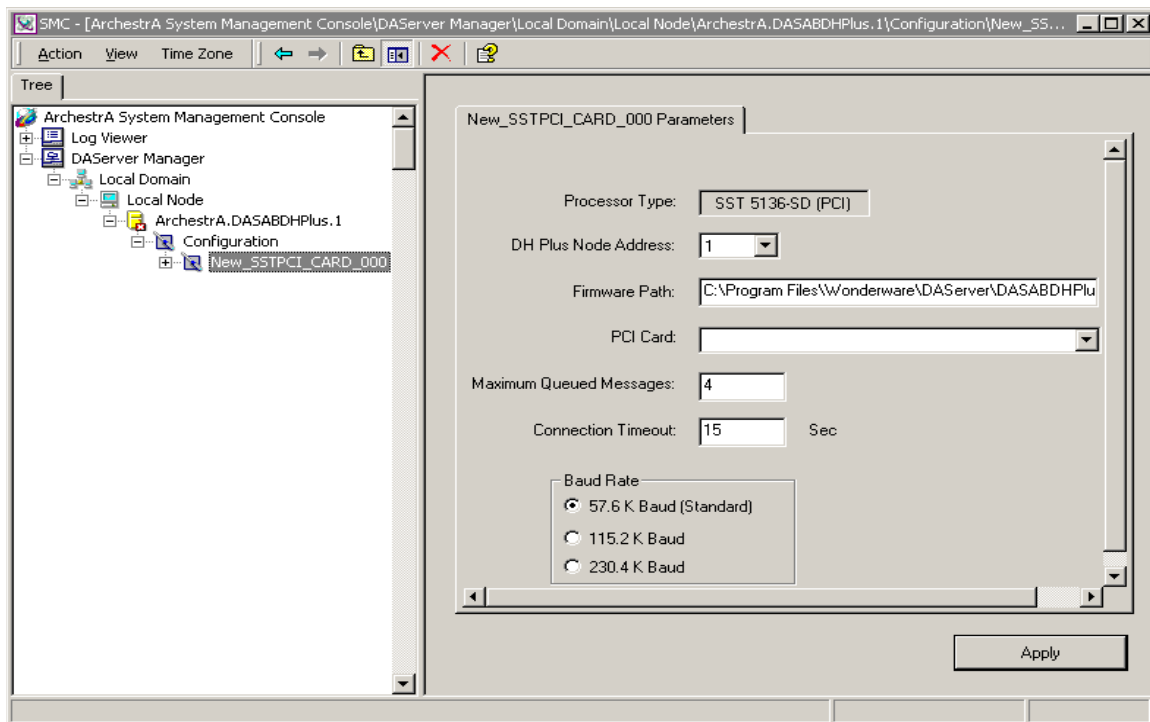
The New_PKT_CARD_000 Parameters configuration view has eight elements, seven of which are configurable.

- **Processor Type:** The information is provided automatically by the DAServer Manager.
 - In this case, the processor type is AB 1784-PKTx (PCI).
- **DH Plus Node Address:** Select the unique Node Address on the Data Highway Plus network.
 - The maximum octal number allowed for this address is 77.
 - The minimum value is 0 (zero).
 - The default value is 1 (one).
- **Firmware Path:** Enter the path of the file containing the firmware.
 - The maximum number of characters that can be used is 255.
 - The default path of the file is: C:\Program Files\Wonderware\DA Server\DA S
- **PCI Card:** Select the adapter card from the list.
- **Maximum Queued Messages:** Set the maximum number of messages to be queued.
 - Valid range should be 1 (one) to 4 (four).
 - The default number is 4 (four).
- **Connection Timeout:** Determine the time allowed to establish a connection to a device.
 - The default value is 15 seconds.

- Valid range is 1 (one) to 300 seconds.
- **Baud Rate:** Set the Baud rate either to 57.6 K Baud or 230.4 K Baud.
 - The standard or default rate is 57.6 K Baud.
- **Terminating Resistor Enabled:** Enable the terminating resistor when the adapter is the first or the last physical station on the Data Highway Plus.

SSTPCI_CARD Object

The configuration of the SST 5136-SD-PCI-type cards takes place in the **New_SSTPCI_CARD_000 Parameters** configuration view. A maximum of four (4) cards can be added to the configuration.



There are six elements in this **New_SSTPCI_CARD_000 Parameters** configuration view that are configurable.

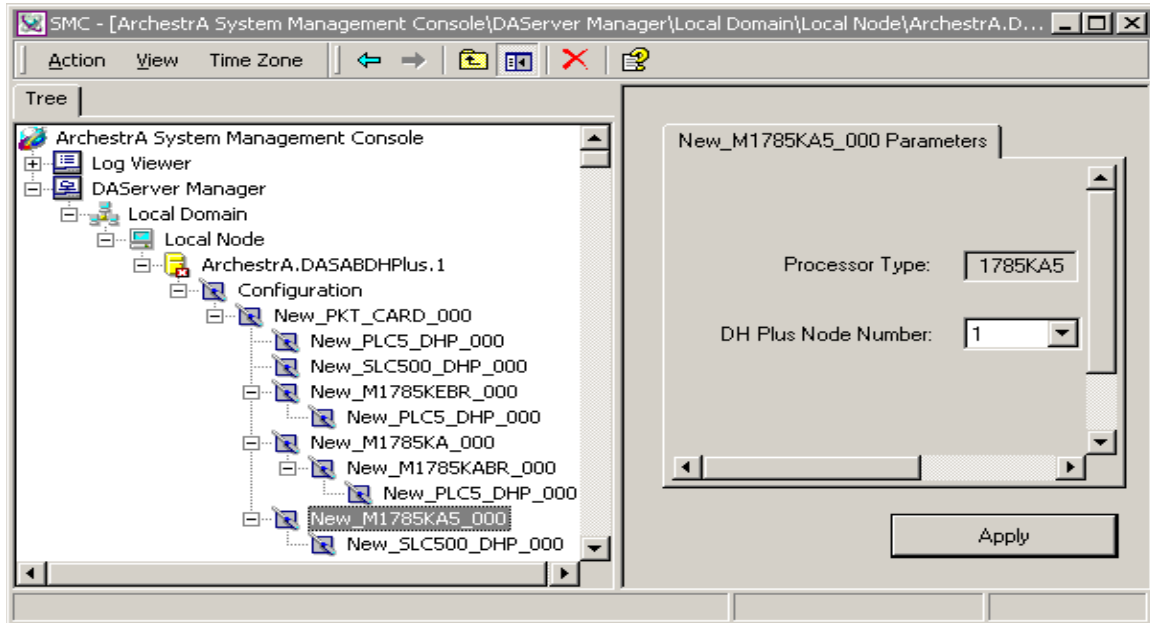
- **Processor Type:** The DA Server Manager automatically provides the information.
 - In this case, the processor type is SST 5136-SD (PCI).
- **DH Plus Node Address:** Select the unique Node Address on the Data Highway Plus network.
 - The maximum octal number allowed for this address is 77.
 - The minimum value is 0 (zero).
 - The default value is 1 (one).
- **Firmware Path:** Provide the path of the file containing the firmware.

- The default file location is C:\Program Files\Wonderware\DA Server\ DASABDHPLUS\Bin\Firmware.
- The maximum number of characters allowed is 255.
- **PCI Card:** Select the adapter card from the list.
- **Maximum Queued Messages:** Set the maximum number of messages to be queued.
 - The valid range should be 1 (one) to 4 (four).
 - The default number is 4 (four).
- **Connection Timeout:** Determine the time allowed to establish a connection to a device.
 - Valid range is 1 (one) to 300 seconds.
 - The default value is 15 seconds.
- **Baud Rate:** Set the Baud rate to either 57.6 K Baud, or 115.2 K Baud, or 230.4 K Baud..
 - The default rate is 57.6 K Baud.

PLC and Bridge-Module Objects

The following five PLC and Bridge-Module objects can be created from the New PKT_CARD_000 and SSTPCI_CARD_000 branches of the DA Server hierarchy:

- **PLC5_DHP Object** – Multiple PLC5_DHP objects can be created for each <NAME>_CARD object. The DHP object represents the Data Highway Plus used for connecting the PLC-5-family controllers to the adapter card.
- **SLC500_DHP Object** – Multiple SLC500_DHP objects can be created for each <NAME>_CARD Object. The DHP object represents the Data Highway Plus used for connecting the SLC500-family controllers to the adapter card.
- **M1785KEBR Object** – Multiple M1785KEBR objects can be created for each <NAME>_CARD object. Each represents the Allen-Bradley Data-Highway-Plus-to-RS-232-to-Data-Highway-Plus bridge module, 1785-KEBR, which is used for the PLC-5 families of controllers to connect to the adapter card.
- **M1785KA Object** – Multiple M1785KA objects can be created for each <NAME>_CARD object. Each object represents the bridge module 1785-KA, which is connecting the PLC-5 families of controllers to the Data Highway Plus to the Data Highway and to another Data Highway Plus having the connection to the adapter card.
- **M1785KA5 Object** – Multiple M1785KA5 objects can be created for each <NAME>_CARD object. Each represents the Allen-Bradley bridge module 1785-KA5 that is utilized for the SLC 500 controllers to connect to the adapter card through DH485 and Data Highway Plus.

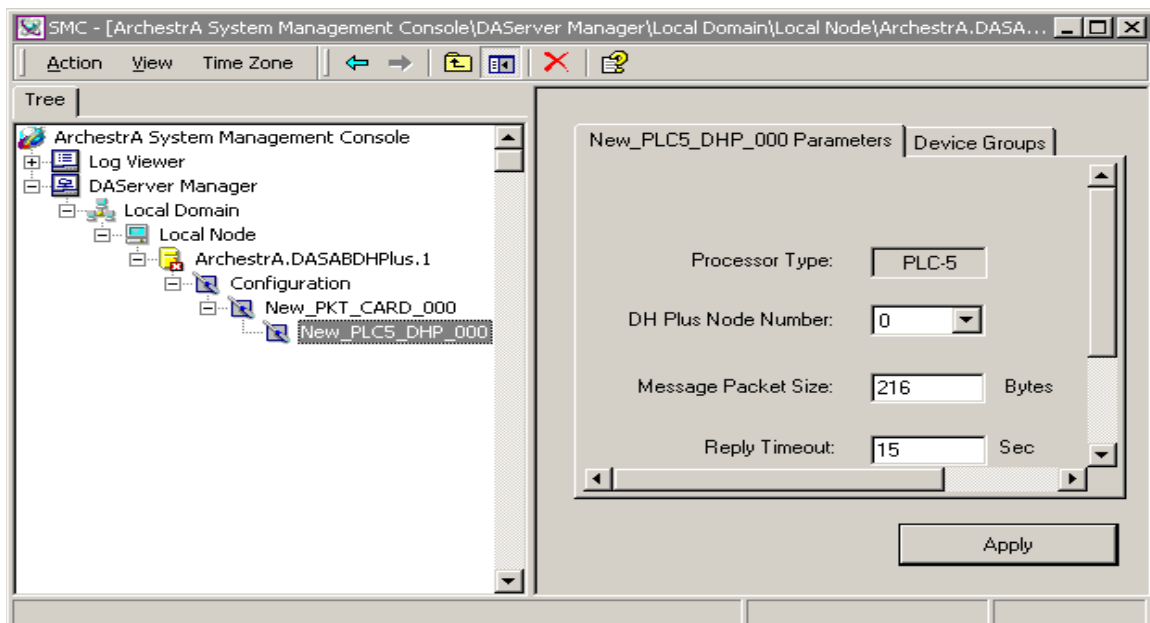


To add PLC and Bridge-Module objects to the hierarchy

1. Right-click on your **New_<NAME>_CARD_000** branch.
2. Select **Add <NAME> Object**.
3. Rename as appropriate.

PLC5_DHP Object

The newly created **PLC5_DHP_000 Parameters** configuration view is the place in the DAServer hierarchy where the Data Highway Plus, used for connecting the PLC-5-family controllers to the adapter card, is configured.

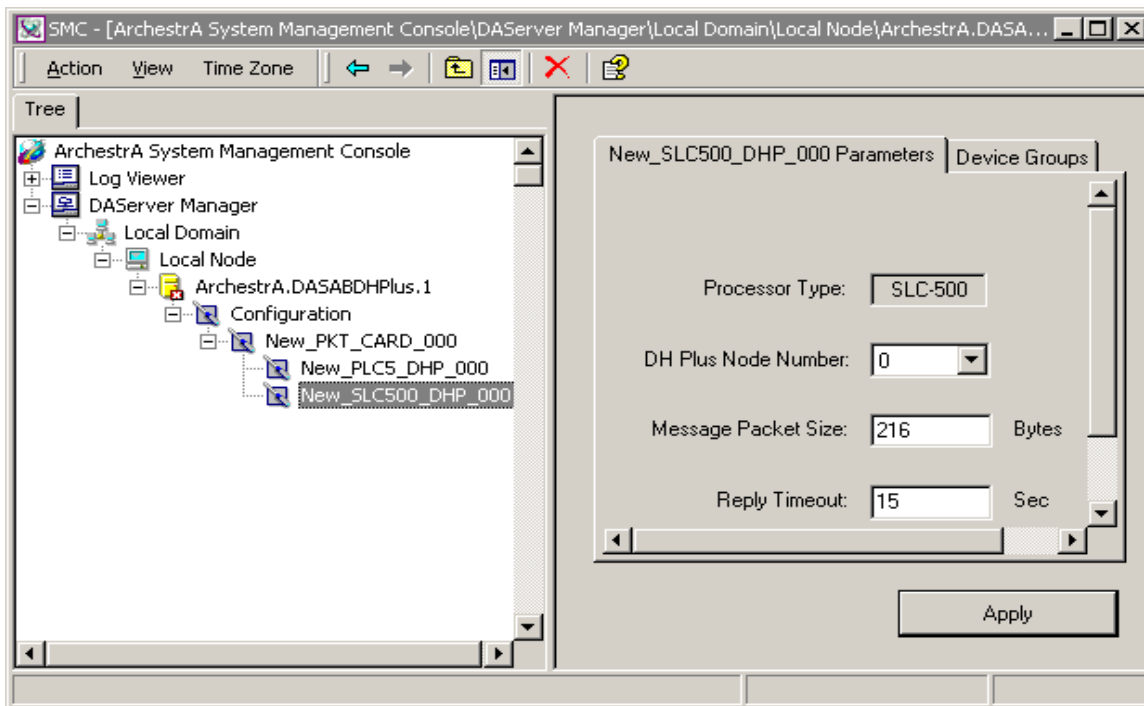


The configuration view contains three configurable parameters:

- **DH Plus Node Number:** Select the Data Highway Plus node number.
 - The maximum octal number allowed for this address is 77.
 - The default value is 1 (one) octal.
- **Message Packet Size:** Determine the message packet size for the PLC-5 controller on the Data Highway Plus.
 - The valid range is 2 (two) to 240 bytes.
 - The default value is 216 bytes.
- **Reply Timeout:** Enter the maximum amount of time that the DAServer will wait for a response from the PLC.
 - The default value is 15 seconds.
 - Valid range is 1 (one) to 300 seconds.

SLC500_DHP Object

The newly created **SLC500_DHP_000 Parameters** configuration view is used to configure the Data Highway Plus used for connecting the SLC500-family controllers to the adapter card.



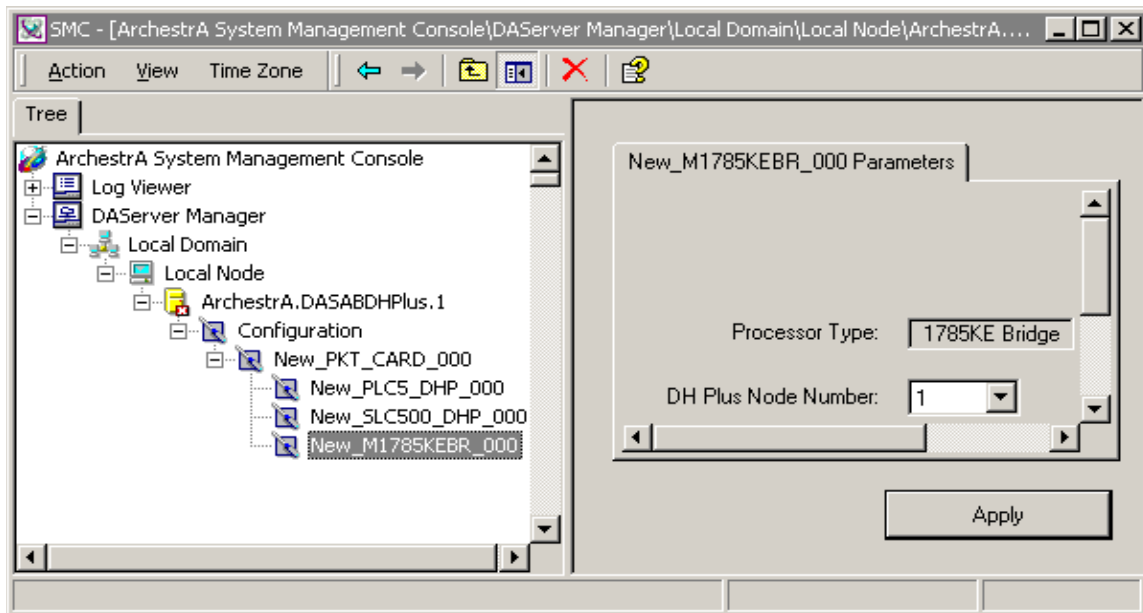
Three configurable elements can be found in the configuration view:

- **DH Plus Node Number:** Select the Data Highway Plus node number.
 - The maximum octal number allowed for this address is 77.
 - The default value is 1 (one) octal.

- **Message Packet Size:** Set the size of the message packet for the SLC 500 controller on the Data Highway Plus.
 - The default value is 216 bytes.
 - Valid range is 2 (two) to 240 bytes.
- **Reply Timeout:** Enter the maximum amount of time that the DAServer will wait for a response from the PLC.
 - The default value is 15 seconds.
 - Valid range is 1 (one) to 300 seconds.

M1785KEBR Object

The **New_M1785KEBR_000 Parameters** configuration view is used to configure the Allen-Bradley 1785-KEBR, a Data-Highway-Plus-to-RS-232-to-Data-Highway-Plus bridge module, which is utilized for the PLC-5 families of controllers to connect to the adapter card.

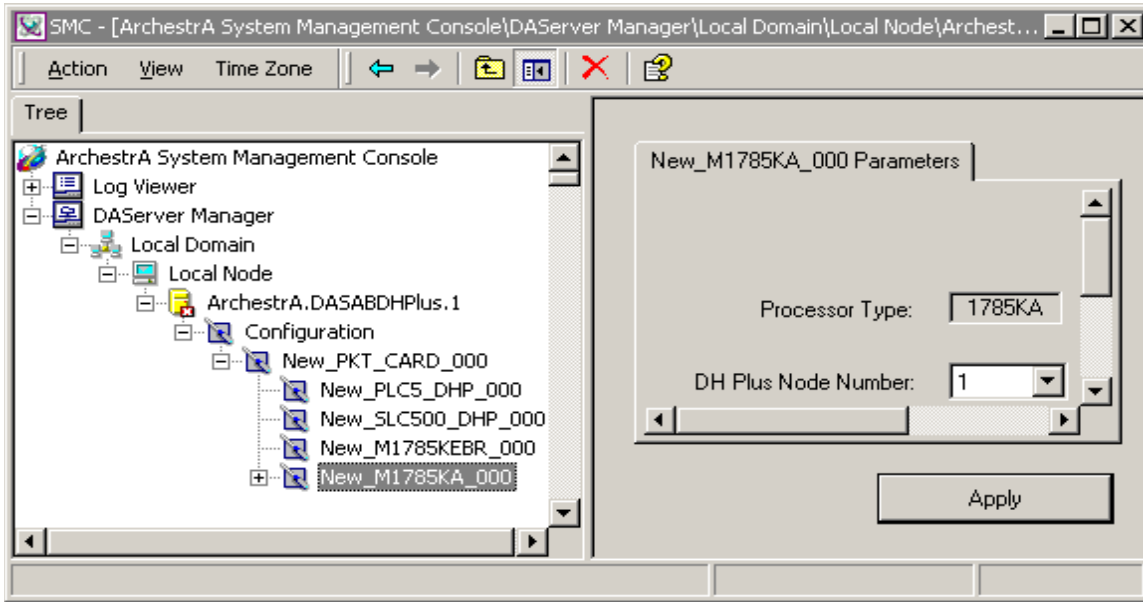


The configuration view contains one configurable element, as follows:

- **DH Plus Node Number:** Select the Data Highway Plus node number.
 - The maximum octal number allowed for this address is 77.
 - The default value is 1 (one) octal.

M1785KA Object

The **New_M1785KA_000 Parameters** configuration view is used to configure the 1785-KA bridge module, which is connecting the PLC-5 families of controllers from the Data Highway Plus to the Data Highway, through M1785-KABR module, and to another Data Highway Plus that has the connection to the adapter card.

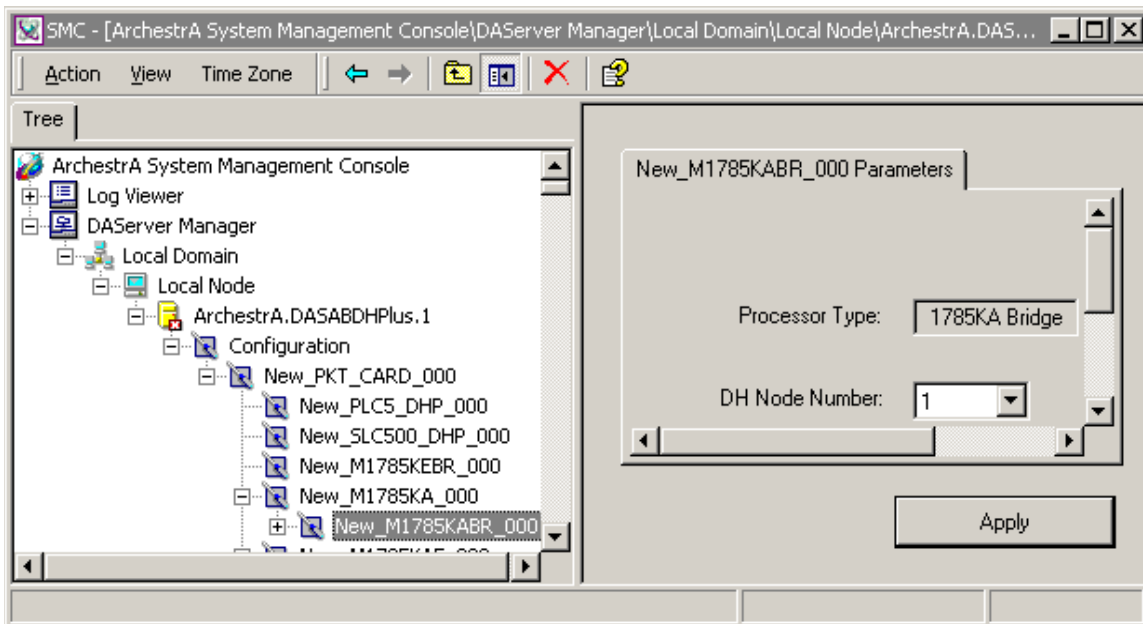


The configuration view contains the following configurable element:

- **DH Plus Node Number:** Select the Data Highway Plus node number.
 - The maximum octal number allowed for this address is 77.
 - The default value is 1 (one) octal.

M1785KABR Object

The **New_M1785KABR_000 Parameters** configuration view is used to configure the 1785-KABR bridge module, which is connecting the PLC-5 families of controllers from the Data Highway Plus to the Data Highway.

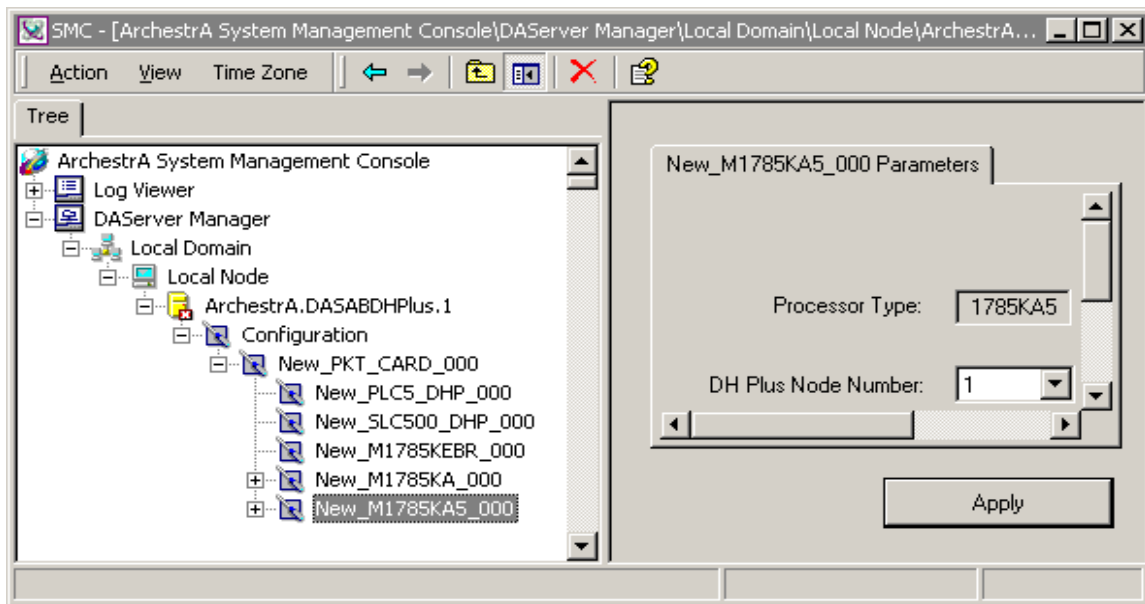


This configuration view contains the following configurable element:

- **DH Node Number:** Select the Data Highway node number.
 - The maximum octal number allowed for this address is 77.
 - The default value is 1 (one) octal.

M1785KA5 Object

The **New_M1785KA5_000 Parameters** configuration view is the place in the DAServer hierarchy to configure the Allen-Bradley bridge module 1785-KA5, which is utilized for the SLC 500 controllers to connect to the adapter card through DH485 and Data Highway Plus.



One element in this configuration view needs to be configured:

- **DH Plus Node Number:** Select the Data Highway Plus node number.
 - The maximum octal number allowed for this address is 77.
 - The default value is 1 (one) octal.

Note The default name created from adding a hierarchy object is in the format of **New_ObjectName_###**, where **ObjectName** is the name of the object type and **###** is a numeric value starting from "000" enumerated sequentially per hierarchy object. The link name for the OPC items is constructed by assembling the respective object names of the nodes along the hierarchy tree in the logical order, starting from this DAServer's CARD Object root down to the leaf. Therefore, the link name is always unique for the DAServer.

Note In order to use the DAServer, you must activate it. See the DAServer Manager documentation for information about how to activate and deactivate the DAServer.

Configuring Device Group Definitions

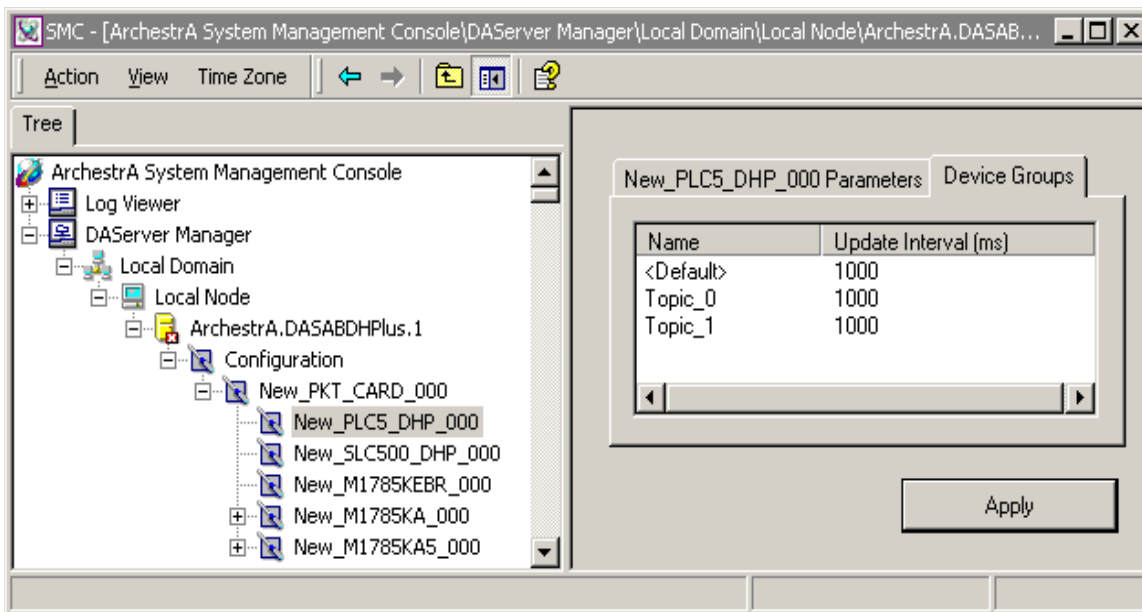
The **Device Groups** tab in the DAServer Manager user interface is used to create new, modify, or delete device group definitions for an object. For DDE/SuiteLink communications, one or more device group definitions must exist for each PLC that the DAServer will communicate with.

Important! For DDE/SuiteLink, it is strongly recommended that each device group (topic) definition contain a unique name. The OPC, however, has the flexibility to use any names, including duplicate names, for the device group definitions.

Device Group Definitions

The **Device Groups** dialog box, which is displayed by clicking the **Device Groups** tab in the PLC5_DHP Object and SLC_500_DHP Object configuration views, is the place to add device groups, delete and define device groups, configure default update intervals for the objects, and also edit update intervals.

Note When you add a new device group, enter a unique name (up to 32 characters long). When you select another part of the DAServer tree hierarchy, you are prompted to save the modifications to the configuration set.



To create or add device groups

1. To create or add device groups, right-click in the **Device Groups** box.
2. Select the **Add** command from the shortcut menu.
 - When you add a new device group, enter a unique name (up to 32 characters long).

To make edits on device groups

The device group's name or update interval for an object can be edited as follows:

- In the **Name** column, double-click on the device group's name to be edited and make the edit.
- Double-click on the device group's value to be edited in the **Update Interval** column, and make the edit.

To delete device groups

- Deleting a device group from the list can be performed by right-clicking on the device group to be deleted, and then selecting the **Delete** command from the shortcut menu.

Note When you select another part of the ABDHPLUS DAServer tree hierarchy, you are prompted to save the modifications to the configuration set.

To configure default update intervals

1. To configure a default update interval for the object, right-click in the **Device Groups** box.
2. Select **Config Default Update Interval** from the shortcut menu.

To edit update intervals

- To edit the update interval for an object, double-click its value in the **Update Interval** column and make the edit.
 - Update Interval is the frequency (in milliseconds) that the DAServer acquires data from the topics associated with that device group.
 - Different topics can be polled at different rates in a PLC by defining multiple device-group names for the same PLC and setting a different Update Interval for each device group.

Note When you select another part of the ABDHPLUS DAServer tree hierarchy, you are prompted to save the modifications to the configuration set.

Each configuration view associated with objects in the DAServer hierarchy tree has a common feature, the **Apply** button.

1. When you modify any **Parameters** or **Device Groups** dialog box, click **Apply** to implement the new modifications.
 - If you do not click **Apply**, the configuration is reset to its original condition (since the last save).
2. After all modifications, you must save when prompted for the new data to be saved to the configuration set.

Scan-Based Message Handling

Wonderware's DAServers are based on the concept of polling a hardware device for information. This polling is driven by a need which is expressed in the form of requests from one or more clients.

For DDE/SuiteLink, once a particular piece of information has been requested by a client, the DAServer formulates its own request and sends that request to the hardware device. The DAServer then waits for a response to its request. Once the information has been received, the DAServer passes that information back to the client, and repeats the process until all clients have ceased requesting information. The rate at which the DAServer will poll a particular device for a specific piece of information is defined in the device group (topic definition) inside the DAServer, using a parameter called the Update Interval. When setting this parameter, there is always a trade-off between the update speed of the device group and the resulting decrease in system responsiveness.

If you use OPC interface, in addition to the capabilities described in the preceding paragraph, the OPC Client also has additional capabilities on Update Interval control.

Since you more than likely want very fast response, the temptation is to set the Update Interval to a value close to 0 (zero) seconds. However, if every point is polled at this rate, the entire system will suffer due to slow response time. Therefore, you should compromise, and set the Update Interval to a more reasonable value. You could also create multiple device groups for each device, setting the Update Interval to different values, then assigning different items to different device groups depending on how quickly the values change and how quickly you want to see an update of those changes.

Some items, like alarms, change very infrequently, but because of their importance require very fast updates. For those kinds of items, you should set the Update Interval at a very small value. If you desire an immediate response, set the Update Interval at 1 (one).

Unsolicited Message Handling

In the world of PLCs and DAServers it is obvious, to even the most casual user, that a PLC will know when a critical event has occurred before the DAServer will have a chance to poll for that data. Therefore, it would seem natural that if a critical event occurs, the PLC should have the capability to inform the DAServer immediately, without having to wait for the DAServer to poll it.

This is the role of an unsolicited message. Once a PLC has determined that a critical condition exists, it can generate a "reply" message to a poll which never occurred. In other words, it can answer the DAServer's question before the DAServer has a chance to ask it.

The ABDHPLUS DAServer supports the following unsolicited messages:

- **Peer-to-Peer:** This method involves DHP-address-to-DHP-address communications.
 - Configure which computer (by its DHP address) receives unsolicited messages in your client application.
 - Two instances of peer-to-peer unsolicited messages are generated by the DAServer:
 - If the value of "Update Interval" for a topic is 0 (zero), the server will poll this topic only once at the start. After that, only an unsolicited message will update the data.

- If the value of "Update Interval" for a topic is >0 (zero), the server will update the data for a particular item immediately upon receiving an unsolicited message for the item. The DAServer will also update the data at every Update Interval.

Note For a more specific Allen Bradley definition of peer-to-peer unsolicited messaging, please refer to the Allen-Bradley PLC-5 Family documentation.

To receive unsolicited messages

- Set a device group's "Update Interval" to 0 (zero).

To access the settings for device groups

1. Click on the PLC's name in the **Configuration** hierarchy of your DAServer.
2. Select the **Device Groups** tab of the configuration view pane at right.
3. Double-click the number in the Update Interval column of the desired device group and type the number 0 (zero).
 - You are prompted to save the configuration change when you access another part of the DAServer hierarchy tree.

Archiving Configuration Sets

After you have configured your DAServer, you can archive that specific configuration. You can archive more than one configuration set, and subsequently choose different configurations for different purposes.

To archive configuration sets

1. In the DAServer Manager, right-click on the **Configuration** node in the hierarchy below your DAServer.
2. Select **Archive Configuration Set** from the shortcut menu.
3. In the **Archive Configuration Set** configuration view, provide a Configuration Set Name and click **Archive**.
 - All current configuration values are saved to the archived set.

Once you have archived at least one configuration set, you can select it for use.

To use different configuration sets from the current one

1. In the DAServer Manager, right-click the **Configuration** node in the hierarchy below your DAServer.
2. Select **Use Another Configuration Set** from the shortcut menu and click on a configuration set in the sub-menu.
 - All parameters in the DAServer configuration hierarchy change to the chosen configuration set.

Hot Configuration

The ABDHPLUS DAServer is hot-configurable. Incorporated in the DAServer are the following hot-configuration functionalities:

- Modifying Global Configuration parameters.
- Adding, deleting, or modifying device nodes (without affecting any other device nodes, excluding the children of the modified device nodes).
- Adding, deleting, or modifying device groups.

Limited support is provided for the hot configuration for the server-specific configuration parameters in this release. You can modify server-specific parameters while the server is active. However, to have those changes take effect, you have to restart the DAServer.

Note If changes are made to server-specific parameters while the server is active, the DAServer will issue a warning message to the logger.

CHAPTER 3

Item Names

The Wonderware ABDHPLUS DAServer currently supports item names that follow the conventions described for PLC-5-series and SLC 500-series controllers. This chapter describes the item naming conventions for these devices.

Contents

- PLC-5 Item Naming
- SLC 500 Item Naming
- DAServer Standard System Items
- Generic OPC Syntax

PLC-5 Item Naming

The general format of item names for data from PLC-5 controllers matches the naming convention used by the programming software. In the following format, the parts of the name shown in square brackets ([]) are optional.

[*\$*] X [*file*] : *element* [*.field*] [*/bit*]

\$ – Purely optional.

X – Identifies the file type.

file # – File number (0 - 999 decimal).

File 0 must be Output.

File 1 must be Input.

File 2 must be Status.

element – Element number within the file.

For Input and Output files it must be 0 - 777 octal.

For all other file types, it must be 0 - 999 decimal.

.field – Valid only for Counter, Timer, ASCII String, PID, SFC Status, Block Transfer, and Control files. Refer to the preceding table.

/bit – Valid for all file types except ASCII String and Floating Point.

For Input and Output files it must be 0 - 17 octal.

For all other file types it must be 0 - 15 decimal.

The following table summarizes the valid file types, the default file number for each type, and the fields allowed (if any):

X	File Type	Default File #	.fields
O	Output	0	N/A
I	Input	1	N/A
S	Status	2	N/A
B	Binary	3	N/A
T	Timer	4	.PRE .ACC .EN .TT .DN
C	Counter	5	.PRE .ACC .CU .CD .DN .OV .UN
R	Control	6	.LEN .POS .EN .EU .DN .EM .ER .UL .IN .FD
N	Integer	7	N/A
F	Floating Point	8	N/A
A	ASCII	None	N/A
D	BCD	None	N/A
ST	ASCII String*	None	.LEN
PD	PID*	None	.ADRF .ADRE .BIAS .CA .CL .CT .DB .DO .DVDB .DVN .DVNA .DVP .DVPA .EN .ERR .EWD .INI .KD .KI .KP .MAXI .MAXO .MAXS .MINI .MINO .MINS .MO .OLH .OLL .OUT .PE .PV .PVDB .PVH .PVHA .PVL .PVLA .PVT .SO .SP .SPOR .SWM .TIE .UPD
SC	SFC Status*	None	.DN .ER .FS .LS .OV .PRE .SA .TIM
BT	Block Transfer* (Read Only)	None	.EN .ST .DN .ER .CO .EW .NR .RW .TO .RLEN .DLEN .FILE .ELEM

* Available only on certain PLC-5 models. Check the Processor Manual for the model being used.

Output File Items

O[n]:rg[/b]	n represents the file number, which is optional. If specified, it must be 0 (zero).
	r indicates the rack number (0 - 27 octal).
	g indicates the I/O group (0 - 7 octal).
	b specifies the bit (0 - 17 octal). /b may be omitted if necessary to treat the I/O group as a numeric value.

Examples:

O0:00/0

\$O:177/17

O:3 4BCD (for 16-bit 7-segment display)

Input File Items

I[n]:rg[/b]	n represents the file number and is optional. If specified, it must be 1.
	r indicates the rack number (0 - 27 octal).
	g indicates the I/O group (0 - 7 octal).
	b specifies the bit (0 - 17 octal). /b may be omitted, if necessary, to treat the I/O group as a numeric value.

Examples:

I1:0/0

I:177/17

I:3 4BCD (for 16-bit thumbwheel input)

Status File Items

S[n]:e[/b]	n represents the file number and is optional. If specified, it must be 2.
	e indicates the element number in the file.
	b is optional. If specified, it indicates the bit (0 - 15 decimal).

Note Refer to the 1785 PLC-5 Family Processor Manual (Allen-Bradley Publication 1785-6.8.2) for a complete description of the status file information.

Examples:

\$\$:18 (year)

\$\$2:18 (year)

S2:19 (month)

S2:10/0 (battery low status bit)

Binary File Items

B[n]:e[/b] or B[n]/m	n represents the file number and is optional. If not specified, it is assumed to be 3. If specified, the file number must be 3 - 999 decimal.
	e specifies the element (word) number within the Binary file. It must be 0 - 999 decimal.
	b specifies the bit number within the word and is optional. In the first form (where :e is present), the bit number must be 0 - 15 decimal.
	m specifies the bit number within the file. However, in the second form, no word numbers are specified and the bit number may be 0 - 15999.

Examples:

B:33

B:6/4 (same bit as B/100)

B:3/15999 (same bit as B:999/15)

Timer File Items

T[n]:e[.f]/[b]	n represents the file number and is optional. If not specified, it is assumed to be 4. If specified, the file number must be 3 - 999 decimal.
	e specifies the element number (three words per element) within the Timer file. It must be 0 - 999 decimal.
	.f identifies one of the valid Timer fields. The valid fields for Timer Files are listed in the table. If .f is omitted, it is assumed to be the word containing the status bits.
	b is optional and is normally not used. All of the fields of a timer can be accessed by specifying the .f fields. However, it is possible to use /b to single out a bit in the .PRE or .ACC fields (which are words). For Timer files, the bit number must be 0 - 15 decimal.

Examples:

T4:0.ACC

T4:0.DN

T4:1.PRE

Counter File Items

C[n]:e.f[/b]	n represents the file number and is optional. If not specified, it is assumed to be 5. If specified, the file number must be 3 - 999 decimal.
	e specifies the element number (three words per element) within the Counter file. It must be 0 - 999 decimal.
	.f identifies one of the valid Counter fields. The valid fields for the Counter files are listed in the table. If .f is omitted, it is assumed to be the word containing the status bits.
	b is optional and is normally not used. Specifying the .f fields can access all of the fields of a counter. However, it is possible to use /b to single out a bit in the .PRE or .ACC fields (which are words). For Counter files, the bit number must be 0 - 15 decimal.

Examples:

C5:0.ACC

C5:3.OV

C5:1.PRE

Control File Items

R[n]:e.f[/b]	n represents the file number and is optional. If not specified, it is assumed to be 6. If specified, the file number must be 3 - 999 decimal.
	e specifies the element number (three words per element) within the Control file. It must be 3 - 999 decimal.
	.f identifies one of the valid Control fields. The valid fields for Control files are listed in the table. If .f is omitted, it is assumed to be the word containing the status bits.
	b is optional and is normally not used. Specifying the .f fields can access all of the fields of a Control file. However, it is possible to use /b to single out a bit in the .LEN or .POS fields (which are words). If specified, it indicates the bit (0 - 15 decimal).

Examples:

R6:0.LEN

R6:3.EM

R6:1.POS

Integer File Items

N[n]:e[/b]	n represents the file number and is optional. If not specified, it is assumed to be 7. If specified, the file number must be 3 - 999 decimal.
	e specifies the element number within the Integer file. It must be 0 - 999 decimal.
	b is optional. If specified, it indicates the bit (0 - 15 decimal).

Examples:

N7:0

N7:0/15

N7:3

Floating Point File Items

F[n]:e	n represents the file number and is optional. If not specified, it is assumed to be 8. If specified, the file number must be 3 - 999 decimal.
	e specifies the element number within the Floating Point file. It must be 0 - 999 decimal.

Examples:

F8:0

F8:3

ASCII File Items

An:e[/b]	n represents the file number (NOT optional) and must be 3 - 999 decimal.
	e specifies the element number within the ASCII file. It must be 0 - 999 decimal. Each element in an ASCII file contains two ASCII characters.
	b is optional. If specified, it indicates the bit (0 - 15 decimal).
An:x-y	x and y also specify element numbers. In this form, the item is an ASCII string occupying element x through element y . Each element contains two ASCII characters: the first character is the high-order byte and the second is the low-order, and so on.

Note If reading only one word as a two-character string, the range must be "x-x." For example, A20:3-3.

Examples:

A20:3

A10:0/0

A9:0-19 (40-character ASCII string)

BCD File Items

Dn:e[/b]	n represents the file number (NOT optional) and must be 3 - 999 decimal.
	e specifies the element number within the BCD file. It must be 0 - 999 decimal. Each element in a BCD file contains a number between 0 - 9999.
	b is optional. If specified, it indicates the bit (0 - 15 decimal).

Examples:

D20:3

D10:0/3

ASCII String Section Items

STn:e[.f]	n represents the file number (NOT optional) and must be 3- 999 decimal.
	e specifies the element number within the String file. It must be 0 - 779 decimal. Each element in a String file contains an ASCII string with a maximum length of 82 characters.
	.f identifies the following ASCII string field: .LEN. If .f is omitted, it is assumed to be the string.

Examples:

ST9:0

ST9:700

ST9:700.LEN

Block Transfer Section Items

BTn:e.f[/b]	n represents the file number (NOT optional) and must be 3 - 999 decimal.
	e specifies the element number (three words per element) within the Block Transfer file (0 - 999 decimal).
	.f identifies one of the valid Block Transfer fields. The valid fields for Block Transfer items are listed in the table. If .f is omitted, it is assumed to be the word containing the status bits.
	b is optional and is normally not used. Specifying the .f fields can access all of the fields of a Block Transfer. However, it is possible to use /b to single out a bit in the .FILE or .ELEM fields (which are words). For Block Transfer files, the bit number must be 0 - 15 decimal.

Note Block Transfer files are read only.

Examples:

BT9:0.EN

BT9:3.RLEN

BT9:3.FILE

PID Section Items

PDn:e.f[/b]	n represents the file number (NOT optional) and must be 3 - 999 decimal.
	e specifies the element number within the PID file. It must be 0 - 398 decimal.
	.f identifies one of the valid PID fields. The valid fields for PID files are listed in the table. If PID field .ADDR is needed, use .ADRE for element and .ADRF for file.
	b is optional and is normally not used. All of the fields of a PID can be accessed by specifying the .f fields. If specified, it indicates the bit (0 - 15 decimal).

WARNING! Access to PID files may degrade the DAServer's performance due to the extreme size of the PID element (82 words each). If accessing only a few PIDs at a time, performance will not be greatly affected. If accessing a few fields of many PIDs at once, it may be faster to move the needed fields to an intermediate file (Floating Point or Binary) and let the DAServer access the intermediate files.

Examples:

PD9:2.SP

PD9:3.OLH

PD9:0.INI

SFC Status Section Items

SCn:e.f[/b]	n represents the file number (NOT optional) and must be 3 - 999 decimal.
	e specifies the element number within the SFC Status file. It must be 0 - 999 decimal.
	.f identifies one of the valid SFC fields. The valid fields for SFC files are listed in the table.
	b is optional and is normally not used. Specifying the .f fields can access all of the fields of an SFC. For SFC Status items, the bit number must be 0 - 15 decimal.

Examples:

SC9:0

SC9:0.PRE

SC9:0.SA

SLC 500 Item Naming

The general format of item names for data from SLC 500 controllers matches the naming convention used by the programming software. In the following format, the parts of the name shown in square brackets ([]) are optional.

[**\$**] **X** [**file #**] : **element** [**.field**] [**/bit**]

\$ – Purely optional.

X – Identifies the file type.

file # – Identifies the file number. File number must be 0 - 255 decimal. File 0 must be Output, file 1 must be Input, and file 2 must be Status.

element – Element number within the file. For Input and Output files it must be 0 - 30 decimal. All other file types, it must be 0 - 255 decimal.

.field – Valid only for Counter, Timer, and Control files; see table above.

/bit – Valid for all file types except ASCII String and Floating Point. Must be 0 - 15 decimal.

The following table summarizes the valid file types, the default file number for each type, and the **.fields** allowed (if any):

X	File Type	Default File #	.fields
O	Output*	0	N/A
I	Input*	1	N/A
S	Status	2	N/A
B	Binary	3	N/A
T	Timer	4	.PRE .ACC .EN .TT .DN
C	Counter	5	.PRE .ACC .CU .CD .DN .OV .UN .UA

X	File Type	Default File #	.fields
R	Control	6	.LEN .POS .EN .DN .ER .UL .IN .FD
N	Integer	7	N/A
F	Floating Point*	8	N/A
A	ASCII*	None	N/A
ST	ASCII String*	None	N/A

*Available only on certain SL-500 models. Check the Processor Manual for the model being used. If the Floating Point file type is not supported, file 8 is reserved and unusable.

Output File Items

O[n]:e[/b]	n represents the file number and is optional. If specified, it must be 0 (zero).
	e indicates the element number in the file.
	b specifies the bit (0 - 15 decimal). /b may be omitted if necessary to treat the I/O group as a numeric value.

Note The elements in I/O modules are sequentially mapped into a memory table and are different from the item names in the PLC programming software. Refer to the following "Addressing SLC I/O Modules."

Examples:

O0:0/0

\$O:2/15

O:3 4BCD (for 16-bit 7-segment display)

Input File Items

I[n]:e[/b]	n represents the file number and is optional. If specified, it must be 1.
	e indicates the element number in the file.
	b specifies the bit (0 - 15 decimal). /b may be omitted if necessary to treat the I/O group as a numeric value.

Note The elements in I/O modules are sequentially mapped into a memory table and are different from the item names in the PLC programming software. Refer to the following "Addressing SLC I/O Modules."

Examples:

I1:0/0

I:2/15

I:3 4BCD (for 16-bit thumbwheel input)

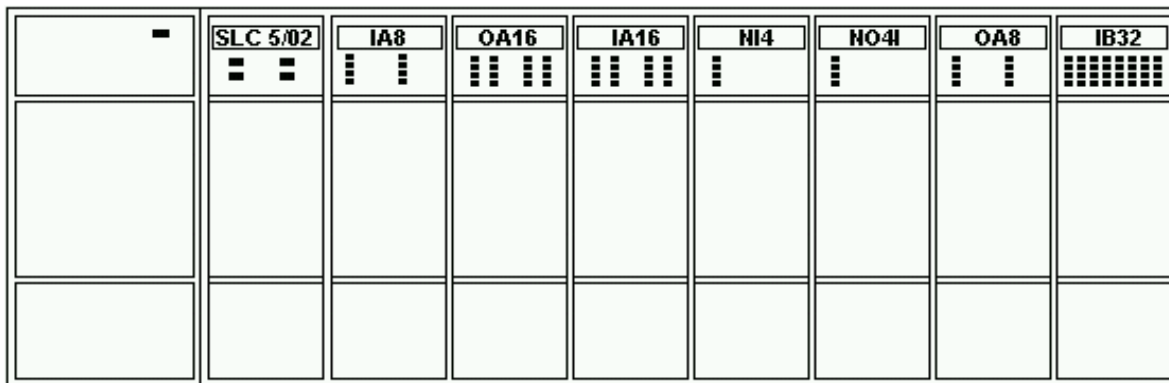
Addressing SLC I/O Modules

The elements (words) in I/O modules are mapped into a memory table. If the Analog I/O modules are being used, then the point naming will differ from the point naming in the programming software. The DAServer item name must be computed from the sum total of words used by the previous input or output blocks. The operator can use the programming software Data Monitor to look at the memory map of the I file or O file to verify your address. If you are not sure of the address, or if the PLC configuration is likely to change, copy the points in question to the N table or B table and access the data from there.

The naming conventions used in the Allen-Bradley programming software are not supported by the Allen-Bradley Ethernet Direct DAServer. The addressing convention is similar to that of the PLC-5 family processors. To derive the correct address for each I/O point, see the following Diagram System. Also see the following topics, Label I/O Modules with "Word Counts," Sequentially Number the Input Modules, and Sequentially Number the Output Modules, to complete addressing the SLC I/O modules.

Diagram System

Addressing of the I/O points begins by drawing a schematic of the system. The following figure is a diagram of the SLC-5/02 system.



The far left unit is the power supply. From left to right, the modules are:

1747-L524	SLC-5/02 Module Processor
1746-IA8	8-point 120VAC input module
1746-OA16	16-point 120VAC output module
1746-IA16	16-point 120VAC input module
1746-NI4	4-point 20mA analog input module
1746-NO4I	4-point 20mA analog output module
1746-OA8	8-point 120VAC input module
1746-IB32	32-point DC input module

Label I/O Modules with "Word Counts"

The address of any point within the I/O datatable space, in an SLC processor, is the sum of the words occupied by previous modules (to the left in the rack) of the same type. Therefore, to determine the correct address for any particular point in the I/O datatable, one must know the number of words each module will consume. Refer to the following list:

Number of Words	Module	
0	1747-L524	SLC-5/02 Module Processor
1	1746-IA8	8-point 120VAC input module
1	1746-OA16	16-point 120VAC output module
1	1746-IA16	16-point 120VAC input module
4	1746-NI4	4-point 20mA analog input module
4	1746-NO4I	4-point 20mA analog output module
1	1746-0A8	8-point 120VAC input module
2	1746-IB32	32-point DC input module

Note In the preceding table, the minimum number of words which can be consumed by a module is 1 (16-bits). This is due to the memory scheme of all Allen-Bradley processors.

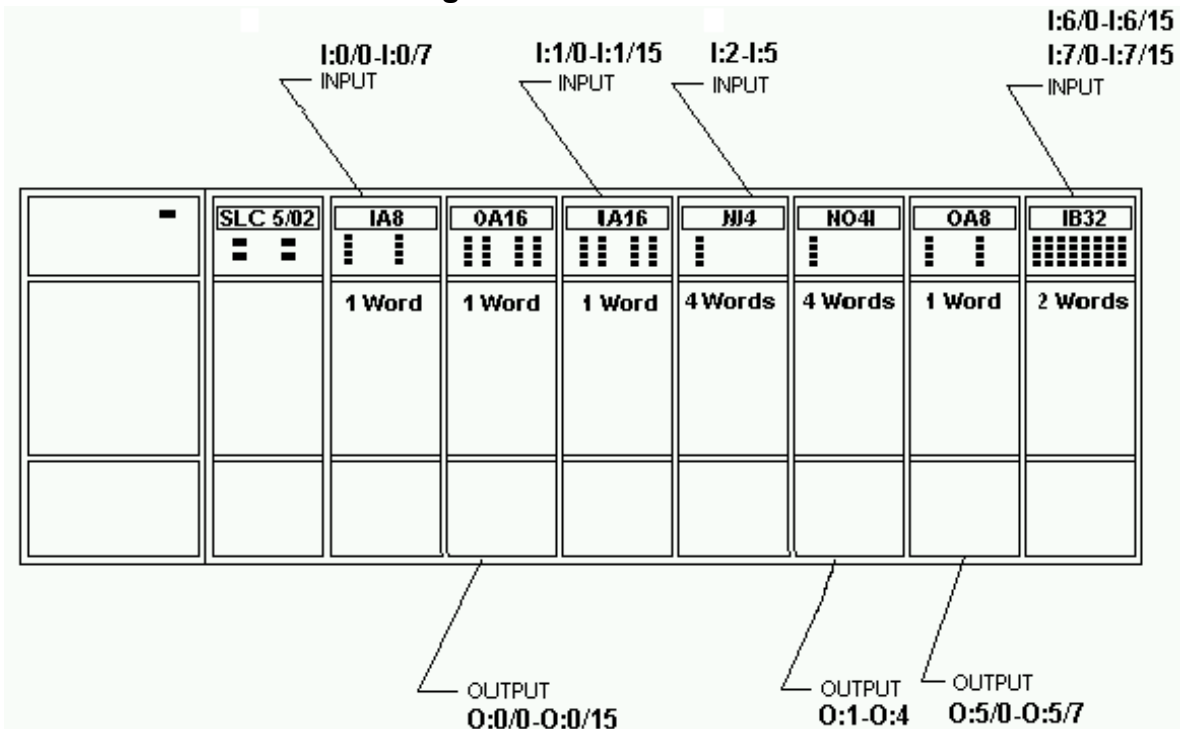
Sequentially Number the Input Modules

In the following I/O diagram, the first input module's addressing should start with "I:0." Previously noted, this module consumes one datatable word. Therefore, the addressing of the next INPUT module encounter, moving from left to right, will begin with "I:1," regardless of the module's physical location.

Sequentially Number the Output Modules

In the following I/O diagram, the first output card encountered is the OA16. Although it is not in the first slot, its address will be "O:0" ("OHH, colon ZERO"). This module consumes one datatable word. Therefore, the addressing of the next OUTPUT module; moving from left to right, will begin with "O:1," regardless of the module's physical location.

I/O Diagram



Status File Items

S[n]:e[b]	n represents the file number and is optional. If specified, it must be 2.
	e indicates the element number in the file.
	b is optional. If specified, it indicates the bit (0 - 15 decimal).

Note Refer to the SLC-500 Family Processor Manual (Allen-Bradley Publication) for a complete description of Status file information.

Examples:

- S2:6 (major error fault)
- S2:13 (math register)
- S:1/5 (forces enabled)

Binary File Items

B[n]:e/b or B[n]/m	n represents the file number and is optional. If not specified, it is assumed to be 3. If specified, the file number must be 3 or 9 - 255 decimal.
	e specifies the element (word) number within the Binary file. It must be 0 - 255 decimal.
	b specifies the bit number within the word. In the first form (where :e is present), the bit number must be 0 - 15 decimal.
	m also represents the bit number. However, in the second form, no word numbers are specified and the bit number may be 0 - 4095.

Examples:

B:33

B:6/4 (same bit as B/100)

B3/4095 (same bit as B:255/15)

Timer File Items

T[n]:e[.f]/[b]	n represents the file number and is optional. If not specified, it is assumed to be 4. If specified, the file number must be 4 or 9 - 255 decimal.
	e specifies the element number (three words per element) within the Timer file. It must be 0 - 255 decimal.
	.f identifies one of the valid Timer fields. The valid fields for Timer Files are listed in the table. If .f is omitted, it is assumed to be the word containing the status bits.
	b is optional and is normally not used. All of the fields of a timer can be accessed by specifying the .f fields. However, it is possible to use /b to single out a bit in the .PRE or .ACC fields (which are words). The bit number must be 0 - 15 decimal.

Examples:

T4:0.ACC

T4:3.DN

T4:1.PRE

Counter File Items

C[n]:e.f[/b]	n represents the file number and is optional. If not specified, it is assumed to be 5. If specified, the file number must be 5 or 9 - 255 decimal.
	e specifies the element number (three words per element) within the Counter file. It must be 0 - 255 decimal.
	.f identifies one of the valid Counter fields. The valid fields for the Counter Files are listed in the table. If .f is omitted, it is assumed to be the word containing the status bits.
	b is optional and is normally not used. Specifying the .f fields can access all of the fields of a counter. However, it is possible to use /b to single out a bit in the .PRE or .ACC fields (which are words). The bit number must be 0 - 15 decimal.

Examples:

C5:0.ACC

C5:3.OV

C5:1.PRE

Control File Items

R[n]:e.f[/b]	n represents the file number and is optional. If not specified, it is assumed to be 6. If specified, the file number must be 6 or 9 - 255 decimal.
	e specifies the element number (three words per element) within the Control file. It must be 0 - 255 decimal.
	.f identifies one of the valid Control fields. The valid fields for the Control files are listed in the table. If .f is omitted, it is assumed to be the word containing the status bits.
	b is optional and is normally not used. All of the fields of a Control file can be accessed by specifying the .f fields. However, it is possible to use /b to single out a bit in the .LEN or .POS fields (which are words). The bit number must be 0 - 15 decimal.

Examples:

R6:0.LEN

R6:3.EN

R6:1.POS

Integer File Items

N[n]:e[/b]	n represents the file number and is optional. If not specified, it is assumed to be 7. If specified, the file number must be 7 or 9 - 255 decimal.
	e specifies the element number within the Integer file. It must be 0 - 255 decimal.
	b is optional. If specified, it indicates the bit (0 - 15 decimal).

Examples:

N7:0

N7:0/15

N7:3

Floating Point File Items

F[n]:e	n represents the file number and is optional. If not specified, it is assumed to be 8. If specified, the file number must be 8 - 255 decimal.
	e specifies the element number within the Floating Point file. It must be 0 - 255 decimal.

Examples:

F8:0

F8:3

ASCII File Items

An:e[/b]	n represents the file number (NOT optional) and must be 9 - 255 decimal.
	e specifies the element number within the ASCII file. It must be 0 - 255 decimal. Each element in an ASCII file contains two ASCII characters.
	b is optional. If specified, it indicates bit (0 - 15 decimal).

Examples:

A20:3

A10:0/0

ASCII String Section Items

STn:e	n represents the file number (NOT optional) and must be 9 - 255 decimal.
	e specifies the element number within the String file. It must be 0 - 255 decimal. Each element in a String file contains an ASCII string with a maximum length of 78 characters.

Examples:

ST9:0

ST9:200

DAServer Standard System Items

System items supply DAServer users with easy access to DAServer status and diagnostics information. They are treated just like ordinary items with respect to the client. However, in most cases these items are not directly acquired via the communications layer. System item values are usually generated through internal calculations, measurements, and the tracking of the DAS Engine.

No DAServer-specific system items are provided in this ABDHPLUS DAServer.

System items, like ordinary items, are defined by the following properties:

- Group (client group/OPC group)
The arbitrary collection of items, not correlated.
- Hierarchical location (link name/OPC path. The hierarchical node section of the fully qualified OPC item ID.)
The device the item is attached to.
- Device group (OPC access path/topic, or a Scan Group on a hierarchical branch.)
The collection of items on the same physical location with the same protocol update rate.

Example:

To check the status of an external device, the reference might be:

ABDHPLUS.ABPLC1.\$SYS\$Status

Note This syntax does not refer to the access path/device group. As long as the data requested is from the same external device, the value will always be the same.

Note For DDE/SuiteLink clients, \$\$SYS\$\$Status always comes from the leaf level of a DAServer hierarchy branch, which is the destination PLC node. For OPC clients, \$\$SYS\$\$Status can be accessed at all hierarchy levels. \$\$SYS\$\$Status at the root level of the whole hierarchy tree is always good, as it represents the quality status of the local computer itself. Hence, for practical application, OPC clients should reference \$\$SYS\$\$Status at any hierarchy levels other than the root.

In the ArchestrA context, the device group plays the most important role of identifying the scope of any item (the device group defines the hierarchical location implicitly when using globally unique device-group names, which is required for DDE/SuiteLink compatibility).

All system items follow the same naming convention:

- All system items start with \$\$SYS\$.
- The DAS Engine scans and parses the name for system items. Parsing of the name is case-insensitive.

All system items can be accessed through subscriptions to a device group. However, while some system items return data for that device group, others are server-wide.

DAServer Global System Item

The following system item refers to specific information regarding a global condition of the DAServer.

System Item Name	Type/ Access Rights	Description	Values
\$\$SYS\$\$Licensed	Boolean/ Read	The Binary status indication of the existence of a valid license for the DAServer. If FALSE, this item causes the DAServer to stop updating existing tags, to refuse activation of new tags, and to reject write requests as well as setting quality for all items to BAD. If TRUE, the DAServer functions as configured. All instances have the same value.	RANGE: 0, 1 1: Valid license exists. 0: No valid license exists.

DAServer Device-Specific System Items

The following system items refer to specific information regarding the device(s) the DAServer is connected to.

System Item Name (Type)	Type/ Access Rights	Description	Values
\$\$SYS\$\$Status	Boolean/ Read	<p>The Binary status indication of the connection state to the device (hierarchy level) the item is attached to.</p> <p>The device group (OPC access path/topic) does not affect the value. The status can be good even if individual items have errors.</p> <p>For DDE/SuiteLink clients, \$\$SYS\$\$Status always comes from the leaf level of a DAServer hierarchy branch, which is the destination PLC node.</p> <p>For OPC clients, \$\$SYS\$\$Status can be accessed at all hierarchy levels.</p> <p>\$\$SYS\$\$Status at the root level of the whole hierarchy tree is always good, as it represents the quality status of the local computer itself. Hence, for practical application, OPC clients should reference \$\$SYS\$\$Status at any hierarchy levels other than the root.</p>	<p>RANGE: 0, 1</p> <p>1: DAServer connection to the device is intact.</p> <p>0: Error communicating with the device.</p>
\$\$SYS\$\$ErrorCode	Longint/ Read	<p>Detailed error code of the communications state to the device.</p> <p>The device group (OPC access path/topic) does not affect the value.</p>	<p>>= 0: Good status (0 is the default state – connected).</p> <p>>0: is some device state such as, connecting, initializing, and so on.</p> <p><0: Error status (value indicates the error).</p>
\$\$SYS\$\$ErrorText	String/ Read	<p>Detailed error string of the communications state of the device.</p> <p>The device group (OPC access path/topic) does not affect the value.</p>	<p>Descriptive text for the communications state corresponding to the error code.</p>

System Item Name (Type)	Type/ Access Rights	Description	Values
\$\$SYS\$StoreSettings	Integer/ Read/Write	Used to make the temporary update interval changes via the \$\$SYS\$updateInterval item permanent. If the client pokes a value of 1 into this system item, the currently set update interval is written to the server's configuration file. The value of this system item clears to 0 after being set, if the configuration file write is successful. If the write fails, then the value is set to -1. If the update interval has been changed via the \$\$SYS\$updateInterval item and this item is not poked to 1, the DAServer uses the original update interval for that topic the next time it is started. Reading the item always provides 0. Read/Write values are persisted only if the user sets this system item. The values other than this persist only for the life of the DAServer.	RANGE: -1, 0, 1 -1: Error occurred during saving the configuration file. 0: Read value always if status is OK. 1: Persist settings (cleared immediately).

DAServer Device-Group-Specific System Items

The following system items refer to specific information regarding device groups that have been configured in the DAServer.

System Item Name (Type)	Type/ Access Rights	Description	Values
\$\$SYS\$updateInterval	DWord/ Read/Write	Used to access the currently set update interval. It is the current update interval of the device group in milliseconds. A client can poke new values into this item. The value of zero indicates that no non-system items on that topic are updated (data for these items are not acquired from the device).	Range: 1...2147483647 0: Topic inactive, no items are updated. Data acquisition is stopped. >0: Expected updated interval for the set of all items in the device group.
\$\$SYS\$maxInterval	DWord/ Read	Used to access the currently measured maximum update interval in milliseconds of all items of the corresponding device group. This item is read only. The value of the slowest item is displayed.	Range: 0...2147483647 0: If update interval is 0 or if the status is false. >0: Measured update interval.

System Item Name (Type)	Type/ Access Rights	Description	Values
\$\$SYSS\$WriteComplete	Integer/ Read/Write	Used to access the state of pending Write activities on the corresponding device group. On device group creation (adding items to an OPC group) the value of this system item is initially 1, indicating all Write activities are complete – no pokes are pending. If values are poked into any items of the device group, the value of this item changes to 0, indicating Write activity is currently in progress. If the server has completed all Write activities, the value of this item changes to 1 if all pokes were successful or to -1 if at least one poke has failed. If the value of this item is not zero, the client can poke 1 or -1 to it (poke a 1 to clear errors or a -1 to test a client reaction on Write errors). If the value of this item is zero, it cannot be poked.	Range: -1, 0, 1 1: Write complete (no Writes are pending – initial state). 0: Writes are pending. -1: Writes completed with errors.
\$\$SYSS\$ReadComplete	Integer/ Read/Write	Used to access the state of initial Reads on all items in the corresponding device group. The value is 1 if all active items in a device group have been read at least once. If at least one item in the device group is activated, this item changes to 0. It changes to 1 if all items have been read successfully or to -1 if at least one item has a non-good quality. Poking a 0 to this item resets the internal Read states of all items in this device group. This resets this item to 0. If all items are read again after this poke, this item changes back to 1 or -1.	Range: -1, 0, 1 1: Read complete (all values have been read). 0: Not all values have been read. -1: All values have been read but some have a non-good quality.
\$\$SYSS\$ItemCount	DWord/ Read	Used to access the number of items in the corresponding device group. This item is read-only.	Range: 0...2147483647 >=0: Number of active items.
\$\$SYSS\$ActiveItemCount	DWord/ Read	Used to access the number of active items in the corresponding device group. This item is read-only.	Range: 0...2147483647 >=0: Number of active items.

System Item Name (Type)	Type/ Access Rights	Description	Values
\$SY\$ErrorCount	DWord/ Read	Used to access the number of all items (active and inactive) that have errors (non-good OPC quality) in the corresponding topic. If the communications status of a device group is bad, all items have errors. This item is read-only.	Range: 0...2147483647 >=0: Number of all items (active and inactive) with errors.
\$SY\$PollNow	Boolean/ Read/Write	Poking a 1 to this item forces all items in the corresponding device group to be read immediately (all messages in this device group become due). This is useful if you want to force to get the newest values from the device, regardless of its update interval. This also works on device groups with a zero update interval (manual protocol triggering).	Range: 0, 1

Generic OPC Syntax

A DAServer serves as a container for OPC Groups, which provide the mechanism for containing and logically organizing OPC items. Within each OPC Group, an OPC-compliant client can register OPC items, which represent connections to data sources in the field device. In other words, all access to OPC items is maintained through the OPC Group.

The fully qualified name for an OPC item is called the Item ID (equivalent to Item Name). The syntax for specifying a unique Item ID is DAServer-dependent. In OPC data acquisition DAServers, the syntax can be as follows:

```
AREA10.VESSEL1.TIC1.PLC.N7:11
```

Where:

Each component (delimited by a period) represents a branch or leaf of the field device's hierarchy.

In this example:

- PLC is the name of the target PLC.
- N7:11 is the specific data point (Item) desired.

An item is typically a single value such as an analog, digital, or string value.

Where Item ID describes the syntax for defining the desired data point, OPC provides for another parameter, called Access Path, that defines optional specifications for obtaining that data. The following table shows the OPC Data Access syntax.

Configuration	Syntax
Adapter -> PLC5/SLC500	<Card name>.<PLC node name>.<Item name>
Adapter -> 1785KA5 -> SLC500	<Card name>.<1785KA5 node name>.<PLC node name>.<Item name>
Adapter -> 1785KE -> PLC5	<Card name>.<1785KE node name>.<PLC node name>.<Item name>
Adapter -> 1785KA1-> 1785KA2 -> PLC5	<Card name>.<1785KA1 node name>.<1785KA2 node name>.<PLC node name>.<Item name>

In DAServers, Access Paths are equivalent to Device Groups; it is this parameter that is used to define the update interval between the DAServer and the field device for accessing the values of data points in the PLC.

CHAPTER 4

Troubleshooting

This chapter describes the troubleshooting tools that can be used to deal with the ABDHPLUS DAServer problems you may encounter.

The DAServer Manager provides access to diagnostics and other statistical data, and the Log Viewer provides access to event messages logged during the operation of a DAServer. Also, your client (for example, InTouch) can monitor connectivity with the PLC through the `$$SYS$$Status` item. Use these tools together with the information in this section to troubleshoot your ABDHPLUS DAServer.

Note In order to determine the version of your DAServer, perform the following steps. Search for `DASABDHPLUS.dll`, right-click on the **File Name**, select **Properties** on the shortcut menu, and select the **Version** tab on the **Properties** dialog box. The version of your DAServer is listed under **File Version**.

Contents

- Monitoring Connectivity Status with the PLC
- Monitoring the Status of DAS Conversations
- Error Messages, Codes, and Conditions

Monitoring Connectivity Status with the PLC

The built-in discrete item, `$$SYS$$Status`, can be used to monitor the status of communications with the PLC. This item is set to the following:

- 0 (zero) when communications with the PLC fails.
- 1 (one) when communications is successful.

Note For DDE/SuiteLink clients, `$$SYS$$Status` always comes from the leaf level of a DAServer hierarchy branch, which is the destination PLC node. For OPC clients, `$$SYS$$Status` can be accessed at all hierarchy levels. `$$SYS$$Status` at the root level of the whole hierarchy tree is always good, as it represents the quality status of the local computer itself. Hence, for practical application, OPC clients should reference `$$SYS$$Status` at any hierarchy levels other than the root.

Enter the following DDE reference formula in the appropriate place in your client:

=DASABDHPLUS|ABPLC!\$SYS\$Status

where:

DASABDHPLUS	is the name of the DAServer application.
ABPLC	is the exact device group defined in the DAServer for the PLC.
\$SYS\$Status	is the discrete item used to monitor the status of connectivity with the PLC.

Enter the following OPC item reference syntax when adding the item in your OPC client:

YourOPCAccessPath.\$SYS\$Status

where:

YourOPCAccessPath	is the assembly of hierarchy node names leading to a specific controller device.
\$SYS\$Status	is the discrete item used to monitor the status of connectivity with the controller device.

Note In the case of a PLC disconnect, the DAServer will retry three times before entering into slow poll mode. In the case of reply time-out, the DAServer will go into slow poll mode immediately.

Monitoring the Status of DAS Conversations

The **InTouch WindowViewer** supports built-in topic names, called **DDEStatus** and **IOStatus**, that can be used to monitor the status of specific DAS conversations. For example, let us assume that **WindowViewer (VIEW)** is communicating with the ABDHPLUS DAServer to a PLC that has been defined in the DAServer with the topic name **ABPLC**. The discrete items, **DDEStatus** and **IOStatus**, are set to 0 when this DAS conversation failed, and to 1 when this DAS conversation is successful.

Using DDEStatus and IOStatus in Excel

The status of communications between the PLC and InTouch can be read into Excel by entering the following DDE reference formula in a cell on a spreadsheet:

=view|DDEStatus!ABPLC

or

=view|IOStatus!ABPLC

where:

view	is the name of the InTouch application.
[DDE][IO] Status	is the built-in topic name used to monitor the status of communications between the DAServer and InTouch.
ABPLC	is the exact topic name defined in the server for the PLC.

Reading Values from the DAServer into Excel

Values may be read directly into Excel spreadsheets from the DAServer by entering a DDE formula into a cell using the following format:

=applicationname|topicname!itemname

Example formula:

=DASABDHPLUS|ABPLC!'N7:0'

where:

DASABDHPLUS	is the name of the DAServer application.
ABPLC	is the exact device group name defined in the DAServer for the PLC.
N7:0	is the actual location in the PLC that contains the data value. This is the item name.

In this example, each time the value of **N7:0** changes in the PLC, the DAServer will automatically send the new value to the cell containing the formula in Excel.

Note Refer to the Microsoft Excel manual for complete details on entering Remote Reference formulas for cells.

Writing Values to the DAServer from Excel

Values may be written to the DAServer from Microsoft Excel by creating an Excel macro that uses the **POKE** command. The proper command is entered in Excel as follows:

```
channel=INITIATE("applicationname","topicname")
=POKE(channel,"itemname", Data_Reference)
=TERMINATE (channel)
=RETURN()
```

The following describes each of the above **POKE** macro statements:

```
channel=INITIATE("applicationname","topicname")
```

- Opens a channel to a specific topic name (defined in the DAServer) in a particular application name (the executable name less the .exe).
- Assigns the number of that opened channel to **channel**.

Note By using the **channel=INITIATE** statement, the word **channel** must be used in the **=POKE** statement instead of the actual cell reference. The **"application name"** and **"topic name"** portions of the formula must be enclosed in quotation marks.

=POKE(channel,"itemname", Data_Reference)

- **POKEs** the value contained in the **Data_Reference** to the specified item name (actual location in the PLC), via the **channel** number returned by the previously executed **INITIATE** function.
- **Data_Reference** is the row/column ID of the cell containing the data value.

=TERMINATE(channel)

- Closes the channel at the end of the macro.
- Some applications have a limited number of channels; therefore, they should be closed when finished.
- **Channel** is the channel number returned by the previously executed **INITIATE** function.

=RETURN()

- Marks the end of the macro.

Note Refer to the **.xlm** sample Excel poke macro provided on the DAServer CD. Also refer to the Microsoft Excel manual for complete details on entering Remote Reference formulas for cells.

Error Messages, Codes, and Conditions

DAServer error messages, controller-specific error messages, ABDHPLUS DAServer error codes, and error conditions generated by the Data Highway Plus are supported. Use the Log Flag data to customize the type of messages logged to the Log Viewer. See the Log Viewer online documentation for more information about using log flags.

To troubleshoot DAServer problems, use the following error messages together with the DAServer Manager Diagnostics root data.

DAServer Error Messages

The following table lists all the generic-DAServer and ABDHPLUS-DAServer-specific error messages.

Error Message	Explanation	Possible Cause	Solution	Log Flag
Unable to read Firmware for card <card name> from <path>	The value of the parameter, " Firmware Path ," is incorrect.	An incorrect path was entered during the configuration.	Deactivate the server and re-enter the correct path in the Firmware Path box of the Configuration dialog box.	DASProtFail
Card failure. Cannot read Adapter card boot_code at addresses 0-3x	The reading of Byte offset into the adapter card's memory failed.	There is a faulty card or a loose contact.	Replace the card.	DASProtFail
Cannot get driver information/status data from AB1784KT.SYS	The reading of the driver info failed.	The driver may not be running.	Restart the server computer.	DASProtFail
The AB1784KT.SYS device driver is not running. GetLastError=<error code> for <card name>	The AB1784KT driver is not running.		Restart the server computer.	DASProtFail
Card failure during RAM test. Timeout waiting for a response. (offset 0x7FF!=1)	The driver verifies the integrity of the RAM used for its internal variable storage, by testing a portion of the RAM after transmission of each packet.		Restart the server computer.	DASProtFail
Card failure of Dual-Port RAM	The integrity of data in the dual-port RAM is protected by the parity error detection in the dual-port hardware.		Restart the server computer.	DASProtFail
Card failure of both CTC Timer and Counter	A problem is encountered in the card's timer or counter.		Restart the server computer.	DASProtFail
Card failure during Serial IO test. Timeout waiting for a response.	There is a problem encountered in the Serial IO of the card.		Restart the server computer.	DASProtFail
Protocol firmware download block too large	An error occurs in the firmware file.	The firmware file may be corrupted.	Replace the firmware file with a new one.	DASProtFail
SS NT kernel device driver not present. See Event Viewer for possible reason	An error occurred during the start-up of the device driver.	The device driver may not be installed properly.	Reinstall the server.	DASProtFail
Board not accepting reset, shows busy	There is a problem in the SST card.		The card may need to be replaced.	DASProtFail

Error Message	Explanation	Possible Cause	Solution	Log Flag
Connection Timeout - Failed to obtain a Queue Box.	Unable to get a free Queue Box.	The Message Send/Receive function is very slow.	Increase the update interval.	DASProtFail
Free Qbox called for Qbox# <number> when packet received but failed to read data buffer	Unable to read data from the received message.		If the server is not able to recover from this, check the network connection.	DASProtFail
Runt (too short) unsolicited message packet received on Adapter Card <card name>	The length of the message is too short.		If the server is not able to recover from this, check the network connection.	DASProtFail
Unsolicited message packet (from node <address>) for unadvised item received on Adapter Card <card name>	Received message has the data for an Item which is not advised.	Client may not have advised the item.	Advise the particular item through client.	DASProtFail
Could not find PLC node when creating message	Failed to find the PLC node from the hierarchy from the message.	When creating the message for the item, the DAServer cannot locate the PLC node. Possible corrupted or bad CFG file.	Check and correct the CFG file.	DASProtFail
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: An address field has an illegal value - check the Topic Configuration dialog, the block sizes may be too large or the PLC type may be incorrect.	When doing this Read or Write operation to the item, the operation failed because an incorrect format in a message was found. The reason for failure is included in the message.	A bad request message was constructed and sent.	Make sure a correctly formatted request message was constructed and sent. Make sure the PLC is configured correctly.	DASProtFail

Error Message	Explanation	Possible Cause	Solution	Log Flag
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Not enough address fields specified - check the Topic Configuration dialog, the block sizes may be too large or the PLC type may be incorrect.</p>	<p>When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.</p>	<p>A bad request message was constructed and sent.</p>	<p>Make sure a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.</p>	<p>DASProtFail</p>
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Too many address fields specified - check the Topic Configuration dialog, the block sizes may be too large or the PLC type may be incorrect</p>	<p>When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.</p>	<p>A bad request message was constructed and sent.</p>	<p>Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.</p>	<p>DASProtFail</p>
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Addressed symbol not found - check the Topic Configuration dialog, the block sizes may be too large or the PLC type may be incorrect.</p>	<p>When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.</p>	<p>A bad request message was constructed and sent.</p>	<p>Make sure a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.</p>	<p>DASProtFail</p>
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Addressed symbol is >8 or <= 0 length - check the Topic Configuration dialog, the block sizes may be too large or the PLC type may be incorrect.</p>	<p>When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.</p>	<p>A bad request message was constructed and sent.</p>	<p>Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.</p>	<p>DASProtFail</p>

Error Message	Explanation	Possible Cause	Solution	Log Flag
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: This address does not exist in PLC - check Data Table definition inside PLC and make sure ALL requested points are defined.	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	A bad request message was constructed and sent.	Make sure a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.	DASProtFail
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: One or more points specified within this message do not exist in the PLC's Data Table - Please define, inside the PLC, the missing point(s).	When doing this Read or Write operation to the item, the operation failed because an incorrect format in the message was found. The reason for failure is included in the message.	A bad request message was constructed and sent.	Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.	DASProtFail
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Cannot complete request; the situation inside the PLC and/or bridge devices has changed since the command started.	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	A bad request message was constructed and sent.	Make sure a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.	DASProtFail
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Data or File is too large - the packet the PLC is trying to send back is too large - check the Topic Configuration dialog.	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	A bad request message was constructed and sent.	Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.	DASProtFail

Error Message	Explanation	Possible Cause	Solution	Log Flag
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Transaction size too large for protocol - check the Topic Configuration dialog, the block sizes may be too large or the PLC type may be incorrect.</p>	<p>When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.</p>	<p>A bad request message was constructed and sent.</p>	<p>Make sure a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.</p>	<p>DASProtFail</p>
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Access Denied; improper privilege - the PLC has refused us access to one or more of the data points in this message - check PRIVILEGE settings inside PLC.</p>	<p>When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.</p>	<p>A bad request message was constructed and sent.</p>	<p>Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.</p>	<p>DASProtFail</p>
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: A resource is not available - A programming terminal may be connected to (communicating with) this PLC - please disconnect the programming terminal.</p>	<p>When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.</p>	<p>Writing to Read/Write operation to a PLC that is already connected to a programming terminal.</p>	<p>Disconnect the programming terminal to the PLC and repeat the Read/Write operation.</p>	<p>DASProtFail</p>
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Condition already exists; resource is already available - a loss of communications synchronization has occurred - if it continues, restart the DAServer.</p>	<p>When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.</p>	<p>A bad request message was constructed and sent.</p>	<p>Make sure that a correctly formatted request message is constructed and sent. If error continues, restart the DAServer and repeat the Read/Write operation.</p>	<p>DASProtFail</p>

Error Message	Explanation	Possible Cause	Solution	Log Flag
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Command cannot be executed - the PLC has refused to execute the command we have sent to it - check the PLC type in the Topic Configuration dialog.	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	A bad request message was constructed and sent.	Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.	DASProtFail
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Histogram Overflow - the PLC's internal stack has overflowed - check PLC settings and hardware for configuration problems	The reason for failure is included in the message.	The PLC setting was incorrect.	Make sure the PLC is configured correctly.	DASProtFail
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: No Access - the PLC has refused us access to one or more of the data points in this message - check PRIVILEGE settings inside the PLC.	When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.	Cannot access the PLC.	Make sure the PLC is configured correctly. Repeat the Read/Write operation after correcting the configuration.	DASProtFail
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: The data type requested (see FileType parameter) may not match the data type for the specified file number inside the PLC.	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	A bad request message was constructed and sent.	Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.	DASProtFail

Error Message	Explanation	Possible Cause	Solution	Log Flag
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: The PLC has rejected the format of our command - check the Topic Configuration dialog, the block sizes may be too large or the PLC type may be incorrect.</p>	<p>When doing this Read or Write operation to the item, the operation failed because the format in the message was found to be incorrect. The reason for failure is included in the message.</p>	<p>A bad request message was constructed and sent.</p>	<p>Make sure a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.</p>	<p>DASProtFail</p>
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: At least one word of the PLC's Data Table we are requesting with this message has been marked for deletion by a programming terminal - this is illegal.</p>	<p>When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.</p>	<p>A bad request message was constructed and sent.</p>	<p>Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.</p>	<p>DASProtFail</p>
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Data conversion error - check PLC configuration and PLC type selection inside the Topic Configuration dialog.</p>	<p>When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.</p>	<p>A bad request message was constructed and sent.</p>	<p>Make sure a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.</p>	<p>DASProtFail</p>
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Remove I/O Scanner not able to communicate with 1771 rack adapter - check PLC for problems.</p>	<p>When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.</p>	<p>A bad request message was constructed and sent.</p>	<p>Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.</p>	<p>DASProtFail</p>

Error Message	Explanation	Possible Cause	Solution	Log Flag
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Remove I/O Adapter cannot communicate with module - check PLC for problems.	When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.	A bad request message was constructed and sent.	Make sure a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.	DASProtFail
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: PLC's remote I/O scanner is unable to update the requested data because of a communication problem - check PLC for problems.	When doing this Read or Write operation to the item, the operation failed because an incorrect format in the message was found. The reason for failure is included in the message.	A bad request message was constructed and sent.	Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.	DASProtFail
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Duplicated Label in Message and/or PLC - you should check PLC setup and Topic Configuration for errors.	When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.	A bad request message was constructed and sent.	Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.	DASProtFail
ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: File is open; another node owns it - a programming terminal may be connected to (communicating with) this PLC - please disconnect the programming terminal.	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	Writing to Read/Write operation is performed to a PLC that is already connected to a programming terminal.	Disconnect the programming terminal to the PLC and repeat the Read/Write operation.	DASProtFail

Error Message	Explanation	Possible Cause	Solution	Log Flag
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] encountered the following error: Another node is the PLC program owner - a programming terminal may be connected to (communicating with) this PLC - please disconnect the programming terminal.</p>	<p>When doing this Read or Write operation to the item, the operation failed because the format in the message was found to be incorrect. The reason for failure is included in the message.</p>	<p>Writing to Read/Write operation is performed to a PLC that is already connected to a programming terminal.</p>	<p>Disconnect the programming terminal to the PLC and repeat the Read/Write operation.</p>	<p>DASProtFail</p>
<p>ERROR - When topic: [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: Unknown Communication Error</p>	<p>When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.</p>	<p>A bad request message was constructed and sent. This is a "catch all" error message.</p>	<p>Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.</p>	<p>DASProtFail</p>
<p>Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Destination node (PLC) is out of buffer space - Therefore, the PLC is refusing to accept our messages</p>	<p>When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.</p>	<p>Cannot communicate with the PLC correctly.</p>	<p>Make sure the PLC is configured and running correctly and successfully. Make sure the PLC is on line and is communicating with the DAServer.</p>	<p>DASProtFail</p>
<p>Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Remote node is not acknowledging (ACK) our requests for data - Is the PLC on-line? Is the bridge device on-line?</p>	<p>When doing this Read or Write operation to the item, the operation failed because an incorrect format in the message was found. The reason for failure is included in the message.</p>	<p>Cannot communicate with the PLC correctly.</p>	<p>Make sure the PLC is configured and running correctly and successfully. Make sure the PLC is on line and is communicating with the DAServer.</p>	<p>DASProtFail</p>

Error Message	Explanation	Possible Cause	Solution	Log Flag
<p>Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Duplicate token holder detected - two stations on the highway have the same address (this is illegal)</p>	<p>When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.</p>	<p>Cannot communicate with the PLC correctly.</p>	<p>Make sure the PLC is configured and running correctly and successfully. Make sure the PLC is on line and is communicating with the DAServer.</p>	<p>DASProtFail</p>
<p>Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Local port is disconnected - we may not be connected (physically) to the highway - check the cabling (wired incorrectly?) and <u>_ALL_</u> connectors (loose?)</p>	<p>When doing this Read or Write operation to the item, the operation failed because the format in the message was found to be incorrect. The reason for failure is included in the message.</p>	<p>Cannot communicate with the PLC correctly.</p>	<p>Make sure the PLC is configured and running correctly and successfully. Make sure the PLC is on line and is communicating with the DAServer. Make sure that cabling is correctly connected.</p>	<p>DASProtFail</p>
<p>Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Application layer timed out waiting for a response</p>	<p>When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.</p>	<p>Cannot communicate with the PLC correctly.</p>	<p>Make sure that the PLC is configured and running correctly and successfully. Make sure that the PLC is on line and is communicating with the DAServer.</p>	<p>DASProtFail</p>
<p>Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Duplicate node detected - two stations on the highway have the same address (this is illegal)</p>	<p>When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.</p>	<p>Cannot communicate with the PLC correctly.</p>	<p>Make sure the PLC is configured and running correctly and successfully. Make sure the PLC is on line and is communicating with the DAServer.</p>	<p>DASProtFail</p>

Error Message	Explanation	Possible Cause	Solution	Log Flag
Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Station is off-line - the PLC we are attempting to contact is not on-line	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	Cannot communicate with the PLC correctly.	Make sure that all PLCs are configured and running correctly and successfully. Make sure the PLC is on line and is communicating with the DAServer.	DASProtFail
Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Hardware fault - a hardware problem has prevented messages from being sent	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	Cannot communicate with the PLC correctly.	Make sure all PLCs are configured and running correctly and successfully. Make sure the PLC is on line and is communicating with the DAServer. Make sure that all hardware is running and functioning correctly.	DASProtFail
Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Unknown Communication Error	When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.	Cannot communicate with the PLC correctly.	Make sure all PLCs are configured and running correctly and successfully. Make sure the PLC is on line and is communicating with the DAServer. Make sure that all hardware is running and functioning correctly.	DASProtFail
Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Illegal command or format - check the Topic Configuration dialog, the block sizes may be too large, or the wrong PLC type may be selected	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	A bad request message was constructed and sent.	Make sure that a correctly formatted request message is constructed and sent. Make sure the PLC is configured correctly.	DASProtFail

Error Message	Explanation	Possible Cause	Solution	Log Flag
Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Remote host will not communicate - PLC may not be on-line	When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.	Cannot communicate with the remote PLC correctly.	Make sure that all PLCs and all remote devices are configured and running correctly and successfully.	DASProtFail
Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Remote host missing or down - PLC may not be on-line	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	Cannot communicate with the remote PLC correctly.	Make sure that all PLCs and all remote devices are configured and running correctly and successfully.	DASProtFail
Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Remote host hardware fault - PLC has experienced a hardware fault	When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.	Cannot communicate with the remote PLC correctly.	Make sure all PLCs and all remote devices are configured and running correctly and successfully. Make sure all hardware is working correctly.	DASProtFail
Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Addressing problem or memory protection rungs (inside the PLC) have prevented us from retrieving the requested information	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	Cannot communicate with the remote PLC correctly.	Make sure that all PLCs and all remote devices are configured and running correctly and successfully. Make sure all hardware is working correctly.	DASProtFail

Error Message	Explanation	Possible Cause	Solution	Log Flag
<p>Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Function disallowed due to command protection inside the PLC, or the position of a hardware key on the PLC</p>	<p>When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.</p>	<p>Cannot communicate with the remote PLC correctly.</p>	<p>Make sure that all PLCs and all remote devices are configured and running correctly and successfully. Make sure all hardware is working correctly.</p>	<p>DASProtFail</p>
<p>Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: PLC processor is in PROGRAM mode - it must be in RUN mode for us to be able to gather data</p>	<p>When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.</p>	<p>Cannot communicate with the remote PLC correctly.</p>	<p>Make sure all PLCs and all remote devices are configured and running correctly and successfully. Make sure that all hardware is working correctly. Make sure that the PLC is not in Program mode; it should be in RUN mode.</p>	<p>DASProtFail</p>
<p>Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Compatibility mode file missing or communication zone problem inside the PLC</p>	<p>When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.</p>	<p>Cannot communicate with the remote PLC correctly.</p>	<p>Make sure that all PLCs and all remote devices are configured and running correctly and successfully. Make sure all hardware is working correctly.</p>	<p>DASProtFail</p>
<p>Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Remote node (PLC or Bridge Device) cannot buffer the command we are trying to send it</p>	<p>When doing this Read or Write operation to the item, the operation failed because the format in the message was incorrect. The reason for failure is included in the message.</p>	<p>Cannot communicate with the remote PLC correctly.</p>	<p>Make sure all PLCs and all remote devices are configured and running correctly and successfully. Make sure all hardware is working correctly. Make sure that the correct message has been constructed to send to the remote PLC.</p>	<p>DASProtFail</p>

Error Message	Explanation	Possible Cause	Solution	Log Flag
Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Remote node (PLC or Bride Device) problem due to download - a download is most likely in progress, please wait until it is complete	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	Cannot communicate with the remote PLC correctly.	Make sure all PLCs and all remote devices are configured and running correctly and successfully. Make sure all hardware is working correctly. Repeat the Read/Write operation after the download operation is completed.	DASProtFail
Error - When topic [TopicName] tried to Read/Write [DESC of the item] it encountered the following error: STS=[sts error code]: Cannot execute command due to active IPBs inside the PLC	When doing this Read or Write operation to the item, the operation failed due to an incorrect format in the message. The reason for failure is included in the message.	Cannot communicate with the remote PLC correctly.	Make sure all PLCs and all remote devices are configured and running correctly and successfully. Make sure all hardware is working correctly.	DASProtFail
A floating point value read for Item: <Item Name> on Node: <Node Name> was NOT A VALID NUMBER (Negative Infinity), therefore, it was converted to +3.4e38	A bad negative floating value was read from the PLC. The value was changed by the DAServer to a known good value before sending it to the client.	Bad data was read.	None.	DASProtFail
A floating point value read for Item: <Item Name> on Node: <Node Name> was NOT A VALID NUMBER (Positive Infinity), therefore, it was converted to +3.4e38	A bad positive floating value was read from the PLC. The DAServer changed the value to a known good value before sending it to the client.	Bad data was read.	None.	DASProtFail
A floating point value read for Item: <Item Name> on Node: <Node Name> was NOT A VALID NUMBER (!NaN!), therefore, it was converted to +3.4e38	A bad positive floating value was read from the PLC. The DAServer changed the value to a known good value before sending it to the client.	Bad data was read.	None.	DASProtFail

Error Message	Explanation	Possible Cause	Solution	Log Flag
Terminating the <Class Name> class - Due to card failed to initialize	Problem occurred during the initialization of the card, which causes the termination of the created object.	Driver may not have started properly. Or, downloading of the firmware may cause the problem.	Check the file path of the firmware in the configuration. Also check other error messages, before this one, for more details.	DASProtFail
The Item in Unsolicited message is not advised: File Type=<Type>, File=<File>, Data Table=<Data>	Received Unsolicited message has data for Item, which is not advised.	Any client does not advise a particular Item. There is a possibility of data loss.	Advise the Item through client.	DASProtFail

PLC-5 Error Messages

The error messages generated specifically for the PLC-5 family controllers are listed in the following table.

Error Message	Explanation	Possible Cause	Solution	Log Flag
Item <ItemName> not valid, PLC does not have PID feature	PID feature is not supported for this PLC-5 configuration.	PLC-5 configuration indicates that the PID feature is not supported.	Select Support PID feature option for PLC-5, if the PLC supports the feature.	DASProtFail
BINARY file number must be greater than 2	Incorrect format for this item. The Binary Item's File Number was smaller than 3 for PLC-5.	The Binary Item's File Number must be 3 or larger.	Only access a Binary Item with a File Number that is equal to 3 or larger.	DASProtFail
TIMER file number must be greater than 2	Incorrect format for the item. The Timer Item's File Number was smaller than 3 for PLC-5.	The Timer Item's File Number must be 3 or larger.	Only access a Timer Item with a File Number equaling to 3 or larger.	DASProtFail
CONTROL file number must be greater than 2	The format for the item is incorrect. The Control Item's File Number was smaller than 3 for PLC-5.	The Control Item's File Number must be 3 or larger.	Only access a Control Item with a File Number that is equal to 3 or larger.	DASProtFail
INTEGER file number must be greater than 2	Incorrect format for the item. The Integer Item's File Number was smaller than 3 for PLC-5.	The Integer Item's File Number must be 3 or larger.	Only access an Integer Item with a File Number that is equal to 3 or larger.	DASProtFail
BCD file number must be greater than 2	The format for the item is incorrect. The BCD Item's File Number was smaller than 3 for PLC-5.	The BCD Item's File Number must be 3 or larger.	Only access a BCD Item with a File Number equaling to 3 or larger.	DASProtFail

Error Message	Explanation	Possible Cause	Solution	Log Flag
PD file number must be greater than 2	Incorrect format for the item. The PID item's File Number was smaller than 3 for PLC-5.	The PID Item's File Number must be 3 or larger.	Only access a PID Item with a File Number that is equal to 3 or larger.	DASProtFail
ST file number must be greater than 2	Incorrect format for the Item. The ST Item's File Number was smaller than 3 for PLC-5.	The ST Item's File Number must be 3 or larger.	Only access an ST Item with a File Number that is equal to 3 or larger.	DASProtFail
BT file number must be greater than 2	Incorrect format for the Item. The BT Item's File Number was smaller than 3 for PLC-5.	The BT Item's File Number must be 3 or larger.	Only access a BT Item with a File Number equaling to 3 or larger.	DASProtFail
SC file number must be greater than 2	The format for the Item is incorrect. The SC Item's File Number was smaller than 3 for PLC-5.	The ST Item's File Number must be 3 or larger.	Only access an ST Item with a File Number that is equal to 3 or larger.	DASProtFail
Cannot write to file BT[FileNumber]	Failed to write to a BT item for PLC-5.	For PLC-5, Write operation to a BT item is not permitted.	Do not attempt to write to a BT item for PLC-5.	DASProtFail

SLC 500 Error Messages

The following table lists all the SLC-500-controller-specific error messages.

Error Message	Explanation	Possible Cause	Solution	Log Flag
BINARY file number must be greater than 8	Incorrect format for this item. The Binary Item's File Number was smaller than 9 for SLC 500.	The Binary Item's File Number must be 9 or larger.	Only access a Binary Item with a valid File Number.	DASProtFail
TIMER file number must be greater than 8	Incorrect format for the item. The Timer Item's File Number was smaller than 9 for SLC 500.	The Timer Item's File Number must be 9 or larger.	Only access a Timer Item with a valid File Number.	DASProtFail
COUNTER file number must be greater than 8	The format for the item is incorrect. The Counter Item's File Number was smaller than 9 for SLC 500.	The Counter Item's File Number must be 9 or larger.	Only access a Counter Item with a valid File Number.	DASProtFail

Error Message	Explanation	Possible Cause	Solution	Log Flag
CONTROL file number must be greater than 8	Incorrect format for the item. The Control Item's File Number was smaller than 9 for SLC 500.	The Control Item's File Number must be 9 or larger.	Only access a Control Item with a valid File Number.	DASProtFail
INTEGER file number must be greater than 8	Incorrect format for the item. The Integer Item's File Number was smaller than 9 for SLC 500.	The Integer Item's File Number must be 9 or larger.	Only access an Integer Item with a valid File Number.	DASProtFail

PLC-5 and SLC 500 Error Messages

The error messages listed in the following table pertain to both the PLC-5 and SLC 500 controllers.

Error Message	Explanation	Possible Cause	Solution	Log Flag
File numbers must be between 0 and 999	For PLC-5 and SLC 500: The format for the item was incorrect. The Item's File Number was out of range.	A bad Item's File Number is used.	Use a valid range for the Item's File Number.	DASProtFail
OUTPUT file number must be 0	For PLC-5 and SLC 500: Incorrect format for the item. The Output Item's File Number was not 0.	The Output Item's File Number must be 0.	Only access an Output Item with a File Number that is equal to 0.	DASProtFail
INPUT file number must be 1	For PLC-5 and SLC 500: Incorrect format for the item. The Input Item's File Number was not 1.	The Input Item's File Number must be 1.	Only access an Input Item with a File Number that is equal to 1.	DASProtFail
STATUS file number must be 2	For PLC-5 and SLC 500: Incorrect format for the item. The Status Item's File Number was not 2.	The Status Item's File Number must be 2.	Only access a Status Item with a File Number equaling to 2.	DASProtFail
Binary file, bit>15 and element>0	For PLC-5 and SLC 500: Incorrect format for the item. The Binary Item contained an element number, but its bit number was larger than 15.	For PLC-5 and SLC 500: The valid format for a PLC-5 Binary Item is: B[FileNumber]:[Element]/[Bit], where bit is from 0 to 15. In this case, the bit field was larger than 15.	Only access a Binary Item with a valid range.	DASProtFail

Error Message	Explanation	Possible Cause	Solution	Log Flag
FLOATING POINT file number must be greater than 2	For PLC-5 and SLC 500: The format for the item is incorrect. The Floating Point Item's File Number was smaller than 3.	The Floating Point Item's File Number must be 3 or larger.	Only access a Floating Point Item with a File Number that is equal to 3 or larger.	DASProtFail
FLOATING POINT file cannot have bit number	For PLC-5 and SLC 500: Incorrect format for the item. The Floating Point Item contained a bit number field.	The Floating Point Item must not contain a bit number field.	Only access a Floating Point Item without a bit number field.	DASProtFail
ASCII file number must be greater than 2	For PLC-5 and SLC 500: Incorrect format for the item. The ASCII Item's File Number was smaller than 3.	The ASCII Item's File Number must be 3 or larger.	Only access an ASCII Item with a File Number that is equal to 3 or larger.	DASProtFail
Unsupported file type [File Type]	For PLC-5 and SLC 500: Incorrect format for the item. An invalid Item type was used.	There are no such Item type names.	Use a valid Item type.	DASProtFail
[Sub-Element] not valid for type [FileType] files	For PLC-5 and SLC 500: Incorrect format for the item. The Sub-element is not valid for this File Type.	There is a wrong Item format with a wrong sub-element type.	Only access a valid item format with a correct sub-element type.	DASProtFail
[Sub-Element] not valid for type [FileType] section	For PLC-5 and SLC 500: Incorrect format for the item. The Sub-element is not valid for this section.	The Item format is wrong.	Use only a valid item format.	DASProtFail
Attempt to write read only item in file [FileNumber] element [Element#] subelement [Sub-Element#] ignored	For PLC-5 and SLC 500: the Write operation failed due to an attempt to write to a read-only item.	An attempt to write to a read-only item caused the failure.	Do not attempt a Write operation to a read-only item.	DASProtFail

ABDHPLUS DAServer Error Codes

There are two server-specific error codes, as shown in the following table. These error codes, along with the error text, can be accessed through System Items.

Error Code	Error Text
-10001	PLC not connected
-10002	PLC timeout

Data Highway Plus Error Conditions

The Data Highway Plus generates error conditions. These error conditions and the server-specific strings are generated by the DAServer to the logger.

Allen Bradley Error Code	Logger Message	Log Flag
DHPERR_DP_FNC (0x20000001)	Dual-port memory functionality test error	DASProtFail
DHPERR_RAM (0x20000002)	Unknown random access memory test error	DASProtFail
DHPERR_RAM (0x20000003)	Failure of Z80 RAM 0	DASProtFail
DHPERR_RAM (0x20000004)	Failure of dual-port RAM	DASProtFail
DHPERR_RAM (0x20000005)	Failure of Z80 RAM 1	DASProtFail
DHPERR_RAM01 (0x20000006)	Failure of both Z80 RAM 0 and RAM 1	DASProtFail
DHPERR_RAM1_DP (0x20000007)	Failure of both RAM 1 and Dual-Port RAM	DASProtFail
DHPERR_CTC (0x20000008)	Unknown counter timer circuit test error	DASProtFail
DHPERR_CTC_TMR (0x20000009)	Failure of CTC timer module	DASProtFail
DHPERR_CTC_CNT (0x2000000A)	Failure of CTC counter module	DASProtFail
DHPERR_CTC_TC (0x2000000B)	Failure of both CTC timer and counter modules	DASProtFail
DHPERR_SIO (0x2000000C)	Unknown serial input output test error	DASProtFail
DHPERR_SIO_INT (0x2000000D)	Failure of CIO channel: no interrupt	DASProtFail
DHPERR_SIO_LOOP (0x2000000E)	Failure of SIO channel A: Loopback failure	DASProtFail
DHPERR_PROT_LOAD (0x2000000F)	Protocol file download error	DASProtFail
DHPERR_LOAD_BLK (0x20000010)	Block too large error	DASProtFail
DHPERR_RAM_FULL (0x20000011)	Z80 RAM too full for next block	DASProtFail
DHPERR_BD_WRITE (0x20000012)	Cannot write to adapter card memory	DASProtFail
DHPERR_OPEN_LOADPCL (0x20000013)	Cannot open file LOADPCL.BIN	DASProtFail
DHPERR_OPEN_KLPCL (0x20000014)	Cannot open file KLPCL.BIN	DASProtFail
DHPERR_OPEN_KLST0 (0x20000015)	Cannot open file KLST0.BIN	DASProtFail
DHPERR_OPEN_KLST1 (0x20000016)	Cannot open file KLST1.BIN	DASProtFail
DHPERR_OPEN_KLST2 (0x20000017)	Cannot open file KLST2.BIN	DASProtFail
DHPERR_OPEN_PROT (0x20000018)	Cannot open protocol file	DASProtFail
TIMEOUT_ERR (0x01)	Timeout error	DASProtFail
CANCELLED_ERR (0x02)	Cancelled error code	DASProtFail

CHAPTER 5

Reference

Contents

- DAServer Architecture
- Component Environments

DAServer Architecture

Note DAServers are supported on Microsoft Windows 2000 and Windows XP only. NetDDE protocol is not supported by the DAServers.

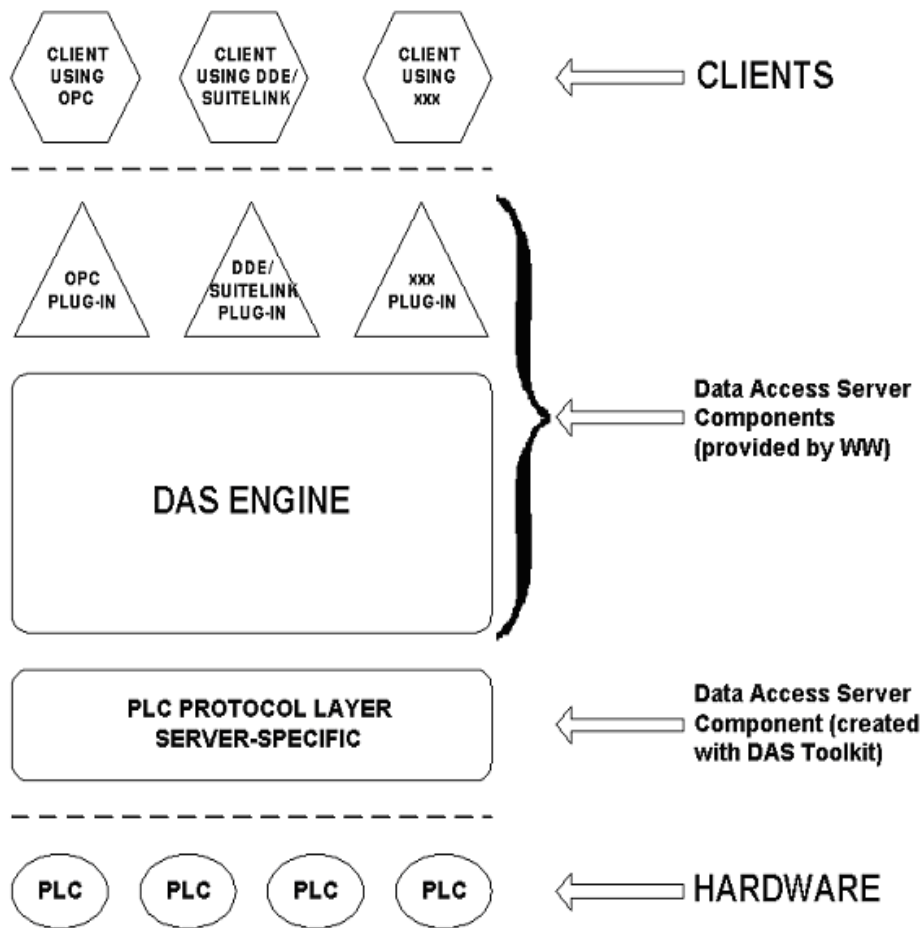
This DAServer is a collection of components that work in concert to provide communications access with hardware field devices. These components include:

- **DAServer Manager:** This is the Microsoft Management Console (MMC) snap-in, which is part of the ArcestrA System Management Console (SMC) suite of utilities, supplied with the DAServer, that provides the necessary user interface for diagnostics, configuration, and activation.
- **Client Plug-ins:** These are the components that are added to a DAServer to enable communications with clients. Examples are: OPC, DDE/SuiteLink, etc.
- **DAS Engine:** This is the library that contains all the common logic to drive data access.
- **Device Protocol:** This is custom code provided by this DAServer to define the communications with a particular device.

DAServers

A DAServer is comprised of three physical parts (see the following figure). They are the following:

- **Plug-in Component(s):** This part is responsible for communicating with clients.
- **DAS Engine:** This common component is used by all DAServers.
- **PLC Protocol Layer, DAServer-specific:** This component is responsible for communicating with the hardware.



DAServer Architecture

Each physical part of a DAServer is comprised of a set of .exe and/or .dll modules. Wonderware provides the plug-ins and the DAS Engine. The DAS Toolkit user creates the PLC Protocol Layer (DAServer-specific) modules. All three sets of modules are required for a fully functioning DAServer.

Plug-ins

Plug-ins provide a protocol translation function for device integration clients. Typical plug-ins communicate in DDE, SuiteLink, or OPC protocol, and serve as interfaces between their clients and the DAS Engine.

Note Items of an array are not supported in the DDE/SL plug-in. These arrays are converted to HEXASCII strings, which provide legacy behavior for DAServers that support this in the DAServer-specific code.

DAS Engine

The DAS Engine is a middleware component that exposes two sets of unique interfaces, one for communicating with the plug-ins and one for communicating with the PLC Protocol Layer components.

PLC Protocol Layer

The PLC Protocol Layer provides a protocol translation function for specific hardware such as ModBus, and serves as an interface between the DAS Engine and the hardware.

Component Environments

Stand-alone DAServers have the following characteristics:

- The DAS Engine is dynamically linked to the other DAServer components. In other words, a new DAS Engine (feature enhancement or bug fix) would not require relinking to the other components nor re-QA of those other components. When deployed to the system, the new DAS Engine would attach to all existing DAServer components.
- Newly deployed Plug-ins (feature enhancements or bug fixes) do not require relinking nor re-QA of associated components. Even new Plug-ins (for example, OPC Alarm & Events) would not require any development changes to the other components, and therefore no relinking in a customer-installed base. In fact, it is feasible to implement new functionality in a plug-in to enhance the DAServer without involvement of the code of the other components.
- DAServers can be configured in one stand-alone configuration utility (DAServer Manager), capable of displaying specific configuration pages for all DAServers. This utility allows browsing and editing of DAServers on different nodes.
- The DAServer Manager diagnostics tool displays generic diagnostic objects common to all DAServers, as well as DAServer-specific/DAServer-developer-defined diagnostic data.

The DAServer data configuration format is XML. Any XML-enabled program (for example, XML Editor) can read this format.

Index

Symbols

\$ 45
 \$\$SYSStatus 61
 .field 37, 45
 /bit 37, 45

A

ABDHPLUS hierarchy 15, 21
 Access Path 58
 Access the settings for device groups 35
 Add PLC and Bridge-Module objects 27
 Application name 13
 ArchestrA System Management Console 20, 85
 ArchestrA.DASABDHPLUS.1 12
 ArchestrA.DASABDHPlus.1 18, 20
 Archive Configuration Set 35
 arrays 86

B

Baud Rate 25, 26
 Before 7
 Boolean item 14
 Bridge Module 11

C

CARD Objects 18, 22
 Client Plug-ins 85
 Communication Protocols 10
 Config Default Update Interval 33
 Configuration 15, 18
 Configuration node 35
 Configuration set 33
 Configuration Set Name 35
 Configure As Service 19, 22
 Configure default update intervals 33
 Configuring the DAServer 20
 Connection Timeout 24, 26
 Create or add device groups 32

D

DAS Engine 85
 DAS Toolkit 86
 DAServer Manager 12, 20, 21, 85
 DAServer Manager book 21
 DAServer Manager documentation 18, 21
 DAServer Manager tree 18
 DAServer version 61
 Data Highway 26, 29, 30
 Data Highway Plus 11, 19, 26
 DCOM 10
 DDE 11
 DDEStatus 62
 Default update interval 33
 Delete device groups 33
 Demo mode 14
 Device group 12, 53

Device Group Definitions 32
 Device Groups 15, 19, 21
 Device Groups box 32
 Device Groups dialog box 32
 Device Groups tab 21, 32
 Device hierarchy 15, 21
 device integration clients 86
 Device Protocol 85
 DH Node Number 31
 DH Plus Node Address 24, 25
 DH Plus Node Number 28, 29, 30, 31
 DH485 26, 31
 Diagnostics 10
 Diagnostics tool 87
 Dynamic Data Exchange 10
 dynamically linked 87

E

Edit mode 18
 Edit update intervals 33
 element 37, 45

F

FastDDE 10
 Features 10, 13
 file # 37, 45
 Firmware Path 24, 25

G

Getting Started Quickly 17
 Group 53
 Group name 12

H

Help menu 21
 Hierarchical location 53
 Hierarchy 22
 HMI 10
 Hot-configuration functionalities 36

I

Index 89
 IOStatus 62
 Item name 12, 13
 Item Names 37

L

License Manager 14
 Link name 12
 Log Flag data 64

M

M1785KA Object 26
 M1785KA object 19
 M1785KA5 Object 26

M1785KA5 object 19
M1785KEBR Object 26
M1785KEBR object 19
Make edits on device groups 33
Manual or automatic service 19, 22
Maximum Queued Messages 24, 26
Message Packet Size 28, 29
Microsoft Management Console 20, 85
middleware component 87
MMC 20, 85
ModBus 87

N

NetDDE 11
Network transport protocol 10
New_M1785KA_000 Parameters 29
New_M1785KA5_000 Parameters 31
New_M1785KABR_000 Parameters 30
New_M1785KEBR_000 Parameters 29
New_PKT_CARD_000 Parameters 23
New_PLC5_DHP_000 Parameters 27
New_SLC500_DHP_000 Parameters 28
New_SSTPCI_CARD_000 Parameters 25
Node name 12, 13

O

OPC 10

P

PCI Adapters 11
PCI Card 24, 26
Peer-to-peer unsolicited messaging 35
PKT_CARD object 18, 23
PLC and Bridge-Module objects 26
PLC disconnect 62
PLC Protocol Layer 85
PLC5_DHP Object 26
PLC5_DHP object 19
Plu-g-in Component 85
Poke command 63
Poke Mode settings 21
Processor Type 24, 25
ProgID 12
Program name 12

R

Receive unsolicited messages 35
reference 85
Remote Reference formulas 64
Reply Timeout 28, 29
Reply time-out 62
Rockwell Software RSLinx 9

S

SCADA 10
Scan Based Message Handling 33
Service 19, 22
Setup.exe 17, 20
SLC500_DHP Object 26
SLC500_DHP object 19

Slow poll mode 62
SMC 20, 85
Snap-in environment 18, 21
SSTPCI_CARD object 18, 23
stand-alone configuration utility 87
Stand-alone DAServers 87
Status file information 39
SuiteLink 10
System Management Console 15, 18, 20

T

Terminating Resistor Enabled 25
To prepare the ABDHPLUS DAServer 17
Topic name 13
Transition mode 21

U

Unsolicited messages 34
Update Interval 34
Update Interval column 33
Use different configuration sets 35

V

Value Time Quality (VTQ) 10

W

WindowViewer 62
WinSock 10
Wonderware folder 18
Wonderware InTouch 9

X

X 45
XML-enabled program 87