

Wonderware DAServer 1.0 for Mitsubishi FX Serial User's Guide

Invensys Systems, Inc.

Revision A

Last Revision: 12/15/06



Copyright

© 2006 Invensys Systems, Inc. All Rights Reserved.

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.

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

<http://www.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

	Welcome.....	7
	Documentation Conventions.....	7
	Technical Support	8
Chapter 1	Getting Started	9
	Before You Begin	10
	Supported Client Protocols	10
	Supported Device Protocols	11
	Supported Devices	11
	Supported Topologies	11
	Windows Firewall Considerations.....	11
	Checklist for Setting up a DAServer	13
	Finding Your DAServer in the SMC	14
Chapter 2	Managing Channels	15
	Adding a Channel.....	15
	Renaming a Channel.....	16
	Setting the Communications Parameters	16
	Setting the ID	17
	Setting the Baud Rate.....	17
	Setting the Data Bits	17
	Setting the Parity	17
	Setting the Stop Bits	17
	Setting Flow Control	17
	Setting Report Communication Errors	18
	Setting the Write/Read Duty Cycle	19
	Deleting a Channel.....	19

Chapter 3	Managing Devices	21
	Adding a Device.....	21
	Renaming a Device.....	22
	Setting the General Parameters.....	22
	Assigning a PLC Model to the Device	22
	Enabling Data Collection.....	23
	Setting the Protocol Parameters	23
	Deleting a Device.....	24
Chapter 4	Managing Device Groups	25
	Adding a Device Group	25
	Renaming a Device Group	26
	Setting Device Group Data	27
	Deleting a Device Group	28
Chapter 5	Managing Device Items	29
	Creating Aliases for Item References.....	30
	Renaming a Device Item	30
	Deleting a Device Item.....	31
	Setting the Item Reference	31
	Exporting and Importing CSV Files.....	32
	Clearing All Device Item Names	33
Chapter 6	Managing Your DAServer	35
	Configuring the DAServer as Service	35
	Configuring the DAServer as Not a Service	36
	Archiving Configuration Sets	36
	Activating/Deactivating the DAServer	38
	In-Proc/Out-of-Proc	39
	Hot Configuration	39
	Demo Mode	39
Chapter 7	Accessing the Data in Your DAServer	41
	Accessing Data Using OPC.....	42
	Accessing Data Using DDE/SuiteLink.....	42
Chapter 8	Item Reference Descriptions.....	43
	Standard System Items	43
	Global System Item	44

Device-Group-Specific System Items	45
Device-Specific System Items	49
Supported Data Types.....	52
Format and Syntax.....	53
String Access.....	53
Overriding the Update Interval or Data Type.....	54
Address Descriptions.....	54
Model FX Address Descriptions.....	54
Model FX0 Address Descriptions.....	56
Model FX0N Address Descriptions.....	57
Model FX0S Address Descriptions	58
Model FX1N Address Descriptions.....	59
Model FX1S Address Descriptions	61
Model FX2N Address Descriptions.....	62
Model FX2NC Address Descriptions	63
Chapter 9 Troubleshooting	65
Troubleshooting with Windows Tools	65
Troubleshooting with the DAServer Manager.....	66
Finding Version Information	66
Using the Wonderware Log Viewer.....	66
Basic Log Flags.....	67
DAServer Log Flags	68
DAServer-Device Interface Log Flags	69
Error Message Descriptions.....	70
Index	73

Welcome

The Wonderware DAServer for Mitsubishi FX Serial is a Microsoft Windows application that allows client applications access to the FX series of Mitsubishi Electric Programmable Logic Controllers (PLC).

Documentation Conventions

This documentation uses the following conventions:

Convention	Used for
Initial Capitals	Paths and filenames.
Bold	Menus, commands, dialog box names, and dialog box options.
Monospace	Code samples and display text.

Technical Support

Wonderware Technical Support offers a variety of support options to answer any questions on Wonderware products and their implementation.

Before you contact Technical Support, refer to the relevant section(s) in this documentation for a possible solution to any problem you have. If you need to contact technical support for help, have the following information ready:

- The type and version of the operating system you are using. For example, Microsoft Windows XP, SP1.
- Details of how to recreate the problem.
- The exact wording of the error messages you saw.
- Any relevant output listing from the Log Viewer or any other diagnostic applications.
- Details of what you did to try to solve the problem(s) and your results.

If known, the Wonderware Technical Support case number assigned to your problem, if this is an ongoing problem.

Chapter 1

Getting Started

The DAServer is one component of a software system that connects your software application with information on the factory floor.

This DAServer documentation covers only the information you need to configure and run the DAServer component; see the documentation that comes with the related components for details on their operation. You can find installation instructions in a help file on the distribution CD.

User interaction with the DAServer is provided through the DAServer Manager. You use the DAServer Manager to configure, activate, and troubleshoot the DAServer. The DAServer Manager is located in the System Management Console (SMC). For more information, see Finding Your DAServer in the SMC on page 14.

This documentation describes some of the features of the DAServer Manager. See the DAServer Manager User's Guide to find more information on:

- Global parameters
- Configuration sets
- Time zone features
- Icon definitions
- Activation/deactivation
- Configuring as a service
- Importing/exporting device items
- Standard diagnostics

You can troubleshoot problems with the DAServer using the Archestra Log Viewer, a snap-in to the SMC. See the Log Viewer help file to find information on:

- Viewing error messages.
- Determining which messages are shown.
- Bookmarking error messages.

You may also be able to troubleshoot problems using your client application, such as the Wonderware InTouch HMI software. The client application can use system device items to determine the status of nodes and the values of some parameters. For more information on system items, see Standard System Items on page 43.

Before You Begin

Before configuring the DAServer, you need to verify the following items:

- A PC is set up with the necessary network cards, and connected to the necessary networks.
- The Windows administration account has been created or identified.
- The DAServer and any other Wonderware software such as the DAServer Manager is installed with the proper licenses. For more information, see the License Utility documentation on the distribution CD.
- The client software is installed.
- The device(s) is/are connected (networked) and, if necessary, programmed.

Before configuring the DAServer, you should know:

- The device network configuration and addresses.
- Which data items are needed for the client application.
- The device name/topic name/group name.
- The desired update intervals.

Supported Client Protocols

Client applications connect to the DAServer using OPC and DDE/SuiteLink. For more information, see the protocols guide that is included with the distribution CD.

Supported Device Protocols

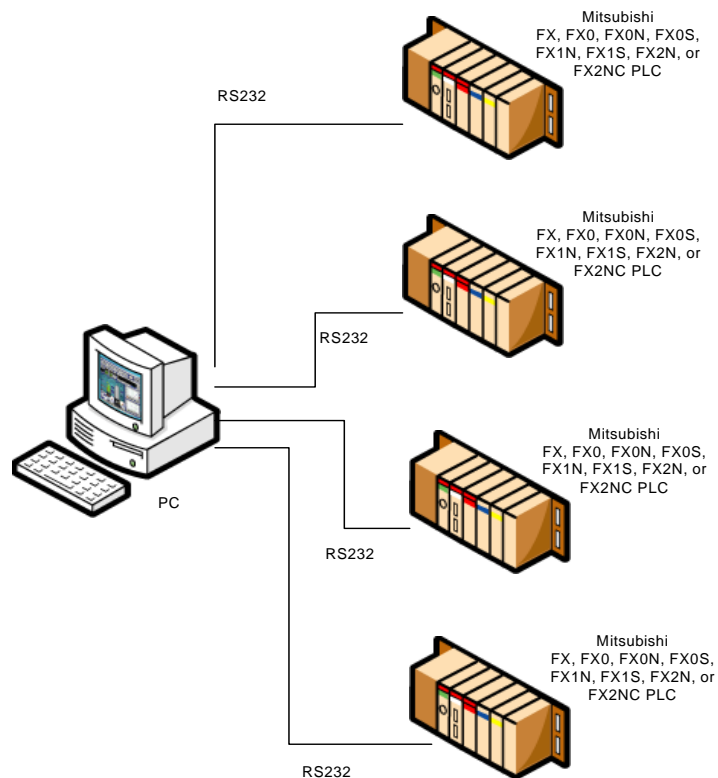
The DAServer supports a direct serial protocol. This protocol does not support simultaneous communication with multiple devices on a single channel.

Supported Devices

You can use Mitsubishi Electric FX, FX0, FX0N, FX0S, FX1N, FX1S, FX2N, and FX2NC devices.

Supported Topologies

This DAServer supports a star network. Each device must be directly connected to the PC running the DAServer. Each serial port can support a single device.



Windows Firewall Considerations

If the DAServer runs on a computer with a firewall enabled, a list of application names or port numbers must be put in the firewall exception list so the DAServer can function correctly.

By default, the DAServer installation program makes the required entries in the firewall exception list. If you do not want the installation program to make entries in the firewall exception list, you must add the entries manually. For information on how make entries in the firewall exception list, see your firewall or Windows security documentation.

Whether you let the install process add the entries, or add them manually, the following applications must be put in the firewall exception list on the computer running the DAServer:

- DASMTFXSerial.exe
- aaLogger.exe
- DASAgent.exe
- dllhost.exe
- mmc.exe
- OPCEnum.exe
- Slssvc.exe

Whether you let the install process add the entries, or add them manually, the following port numbers must be put in the firewall exception list on the computer running the DAServer:

- 5413 a TCP port for slssvc.exe
- 445 a TCP port for file and printer sharing
- 135 a TCP port for DCOM

The following applications must be in the firewall exception list on the computer where the DAServer Manager is installed:

- aaLogger.exe
- dllhost.exe
- mmc.exe

The following port numbers must be in the firewall exception list on the computer where the DAServer Manager is installed:

- 445 a TCP port for file and printer sharing
- 135 a TCP port for DCOM

Un-installing the DAServer does not remove the firewall exception list entries. You must delete the firewall exception list entries manually. For more information on how to do this, see your firewall or Windows security documentation.

Checklist for Setting up a DAServer

If you are setting up a DAServer for the first time, perform the following tasks in the order listed:

- 1 Review the items described in the Before you Begin section. See Before You Begin on page 10.
- 2 Locate the DAServer in the System Management Console (SMC). See Finding Your DAServer in the SMC on page 14.
- 3 Configure the global parameters. See the DAServer Manager User's Guide.
- 4 Add a channel. See Adding a Channel on page 15.
- 5 Set the channel parameters. See Setting the Communications Parameters on page 16.
- 6 Add a device. See Adding a Device on page 21.
- 7 Set the device communication parameters. See Setting the General Parameters on page 22.
- 8 Add one or more device groups. See Adding a Device Group on page 25.
- 9 Add device items. See Managing Device Items on page 29.
- 10 Activate the DAServer. See Activating/Deactivating the DAServer on page 38.
- 11 Access data from the client, including specifying device item references. See Accessing the Data in Your DAServer on page 41 and Setting the Item Reference on page 31.
- 12 Troubleshoot any problems. See Troubleshooting on page 65.

Finding Your DAServer in the SMC

Each DAServer is identified by a unique name. The name for the Wonderware DAServer for Mitsubishi FX Serial is: Archestra.DASMTFXSerial.1. On the computer where the DAServer is installed, it can be found in the local node of the default group of the DAServer Manager.

You do not need to install the DAServer Manager on the same computer as the DAServer. When you access the DAServer remotely, you will not find the DAServer node under the local node. You must locate and identify the DAServer on a computer in one of the node groups.

To find the DAServer

- 1 On the **Start** menu, click **Programs**. Navigate to the Wonderware folder that contains the System Management Console and then click **System Management Console**.
- 2 In the **System Management Console**, expand **DAServer Manager**.
- 3 Locate the group with the node **Archestra.DASMTFXSerial.1**

Chapter 2

Managing Channels

Channels are the communications link between the DAServer and devices. You must create a channel node before you can add device nodes. You can create up to 256 channels.

Before you add a channel, you need to organize and define the topology of the devices being connected.

When you add a channel, the DAServer gives the channel a default name. You may change this name to a name that represents how the network is organized. Channel names cannot have spaces.

A channel has eight parameters. You use six of these parameters to specify how the DAServer transmits over the serial line. You use the seventh parameter to turn on and off reporting errors. The last parameter is the write/read duty cycle.

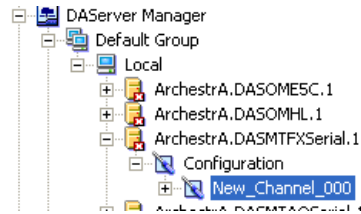
Adding a Channel

The first step in specifying the network between the DAServer and a device is to add a channel. After you add a channel to the hierarchy, you can add device nodes.

To add a channel

- 1 In the DAServer Manager, navigate to the **Configuration** node.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer.

- 2 Right-click **Configuration** and click **Add Channel Object**. The console tree shows a new node with a default channel name selected



- 3 Type a new name and then press Enter.

Renaming a Channel

Changing the channel name prevents clients from registering data using the old name. Data for existing queries is set to bad quality. Try not to make changes to parameters like the channel name after you develop a large client application.

To change an existing channel name

- 1 In the DAServer Manager, navigate to the channel.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer and then expand **Configuration**.
- 2 Right-click the channel and click **Rename**.
- 3 Type the new name and press Enter.

Setting the Communications Parameters

The channel communication parameters are:

- ID
- baud rate
- data bits
- parity
- stop bits
- flow control
- report communication errors

Setting the ID

The ID parameter specifies the port on your computer that the DAServer is to use to communicate with a device. The default value is COM1.

To set the ID

- 1 In the DAServer Manager, navigate to the channel.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer and then expand **Configuration**.
- 2 Select the channel. The right pane shows the channel parameters.



- 3 In the ID box, type or select a COM port.



- 4 Click the **Save** icon.

Setting the Baud Rate

The baud rate is the speed that characters are transmitted between the DAServer and the device. The FX DAServer supports the baud rate value 9600 only.

Setting the Data Bits

The FX DAServer supports data bit value 7 only.

Setting the Parity

The FX DAServer supports parity value “even” only.

Setting the Stop Bits

The FX DAServer supports stop bit value 1 only.

Setting Flow Control

The flow control setting determines how the Request to Send (RTS) and Data Terminal Ready (DTR) control lines are used.

The flow control options are:

- None - no control lines are toggled or asserted.
- DTR - the DTR line is asserted continuously after the communications port is opened.

- **RTS** - specifies that the RTS line will be high if bytes are available for transmission. After all buffered bytes are sent; the RTS line will be low. This is normally used with RS232/RS485 converter hardware.
- **RTS,DTR** - combination of DTR and RTS as described above.
- **RTS Always** - the RTS line is asserted continuously after the communications port is opened.

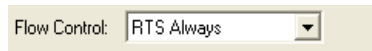
When an RS232/RS485 converter is placed between the DAServer and device, the type of flow control required depends upon the converter. Consult the converter documentation to determine the flow control requirements.

If you use a converter manufacturer's communication cable, you may need to choose a flow control setting of RTS, or RTS always.

The default flow control setting is RTS always.

To set the flow control

- 1 In the DAServer Manager, navigate to the channel.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer and then expand **Configuration**.
- 2 Select the channel. The right pane shows the channel parameters.



- 3 In the **Flow Control** box, type or select a flow option.
- 4 Click the **Save** icon.


Setting Report Communication Errors

You can turn on and off the reporting of low level communications errors. If error reporting is enabled, low-level errors like parity, framing, and overrun are posted to the Logger. If error reporting is disabled, these errors are not posted.

The default is to report communication errors.

To set report communication errors

- 1 In the DAServer Manager, navigate to the channel.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer and then expand **Configuration**.

- 2 Select the channel. The right pane shows the channel parameters.
- 3 Select the **Report Comm. Errors** check box.
-  4 Click the **Save** icon.


Setting the Write/Read Duty Cycle

The duty cycle allows you to control the ratio of write operations to read operations. The ratio is always based on one read for every one to ten writes. The default duty cycle is ten. This means ten writes occur for each read operation.

If your application is doing a large number of continuous writes and you need to ensure that reads are processed in a timely fashion, you may want to reduce the duty cycle. If you set the duty cycle to 1, a single read operation occurs for every write operation. If there are no write operations to perform, reads are processed as needed.

This write/read duty cycle overrides the transaction to subscription ratio global parameter.

To set the write/read duty cycle

- 1 In the DASServer Manager, navigate to the channel.
 - a Expand **DASServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DASServer and then expand **Configuration**.
- 2 Select the channel. The right pane shows the channel parameters.
- 3 In the **Write/Read Duty Cycle** box, type or select a number.
-  4 Click the **Save** icon.

Deleting a Channel

If your computer hardware or the network connection between the computer and devices changes, it may be necessary to delete a channel.

When you delete a channel, all nodes below the channel are also deleted. If a client application requests new data from a deleted channel or from a node on a deleted channel, the request is rejected. Data for existing queries is set to bad quality.

To delete a channel

- 1 In the DASServer Manager, navigate to the channel.

- a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the **DAServer** and then expand **Configuration**.
- 2 Right-click the channel and click **Delete**.
- 3 Read the warning and then click **Yes**. The channel node and all nodes (devices) below it in the hierarchy are deleted.

Chapter 3

Managing Devices

A device communicates with a DAServer and may connect to other devices or I/O points. You set the communication parameters the DAServer uses to communicate with a device.

Adding a Device

You must add a device to the hierarchy before creating device items. A device name cannot contain spaces.

To add a device

- 1 In the DAServer Manager, navigate to the channel node.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer and then expand **Configuration**.
- 2 Right-click the channel node and click **Add Device Object**. The console tree shows a new node with a default device name selected.



- 3 Type a name and press **Enter**.

Renaming a Device

You can change the device name while the DAServer is active. After the device name is changed, client applications using the old name cannot register data with the DAServer. Data for existing queries is set to bad quality. Try not to make changes to the device name after you have developed a large client application.

The device name cannot contain spaces.

To change an existing device name

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Right-click the device whose name you want to change and then click **Rename**.
- 3 Type a name and press Enter.

Setting the General Parameters

The general parameters consist of the device model and whether to enable data collection.

Assigning a PLC Model to the Device

You can only change the model selection if there are currently no client applications connected to the device.

To assign a PLC model to a device

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Parameters** tab. This is shown with the device name followed by the word “Parameters.”



Model:


- 4 In the **Model** list, select the model.
- 5 Click the **Save** icon.

Enabling Data Collection

The enable data collection parameter allows you to control the active state of this device. By default, device communications are enabled.

After a device is disabled, no communications with that device are attempted. In response to client application requests, the quality attribute of the device's data is marked as bad with a sub-status indicating that it is out-of-service and write operations are not accepted. This means that the device is not usable.

To enable data collection

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Parameters** tab. The tab appears with the device name followed by the word "Parameters."
- 4 Select the **Enable Data Collection** check box.
-  5 Click the **Save** icon.

Setting the Protocol Parameters

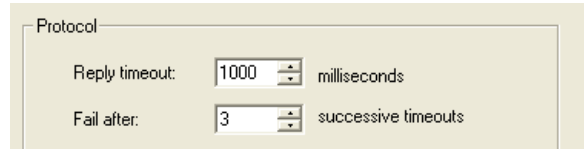
The device protocol parameters consist of the reply timeout and the fail after settings.


- The reply timeout is the time the DAServer waits on a response from the device before giving up and going on to the next request. Longer timeouts only affect performance if a device is not responding.
- The fail after parameter determines how many times the DAServer sends a communications request before the request fails and the device is in error. The valid range is 1 to 10 retries. The default is 3 retries.

To set the communication time out values

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.

- 2 Select the device.
- 3 In the right pane, click the **Parameters** tab. The tab appears with the device name followed by the word “Parameters.”

A screenshot of a configuration window titled "Protocol". It contains two rows of settings. The first row is "Reply timeout:" with a spin box containing the value "1000" and the unit "milliseconds". The second row is "Fail after:" with a spin box containing the value "3" and the unit "successive timeouts".

- 4 In the **Protocol** area, type or select values for the **Reply timeout**, and **Fail after** boxes.
- 5  Click the **Save** icon.

Deleting a Device

Deleting a device removes the node and all device group and device item information. Deleting a device is not reversible. If you make a mistake, you must re-enter the device information. New requests for data that use the device name are rejected. Data for existing queries is set to bad quality.

To delete a device

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Right-click the device and then click **Delete**.
- 3 Read the warning and then click **Yes**.

Chapter 4

Managing Device Groups

Device groups are labels used by client applications when accessing the DAServer. The device group update interval determines how often the DAServer polls the device and sends data to the client application. If you configure multiple device groups with different update intervals, the client application can receive data at various intervals.

Small update intervals mean fast turnaround for data changes and a high overhead because a large amount of data is moving. Large update intervals mean slow turn around for data changes and a low overhead because not as much data is being passed to the client application.

For DDE/SuiteLink clients, the device group is the same as the DDE/SuiteLink topic. DDE/SuiteLink clients require that at least one device group be created for each device.

For OPC clients, the device group equals the OPC access path. The DAServer has a default device group for each device, and this device group cannot be deleted. If you are using OPC client applications, creating a device group is optional.

Adding a Device Group

Device groups allow you to specify an update interval for a set of device items. The device group does not contain any device items. The linkage is made when the client makes a request.

To add a device group

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Device Groups** tab.
- 4 Move the pointer over the columns, right-click, and click **Add**. A **Name** field is selected with a default name.

Name	Update Interval (ms)
Topic_0	1000

- 5 Type a name and press **Enter**. The **Update Interval** field defaults to 1000. The unit is in milliseconds (ms). A device group is added.



- 6 Click the **Save** icon.


Renaming a Device Group

Changing the name of an existing device group requires that any client queries using the device group must be changed. Requests for data accepted by the DAServer before the change are not affected.

To change a device group name

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Device Groups** tab.
- 4 Move the pointer over the name or the box containing the name to be changed. Right-click the name and then click **Rename**. The name is selected.

Name	Update Interval (ms)
Topic_0	1000

- 5 Type a name and press Enter.
- 6  Click the **Save** icon.


Setting Device Group Data

The device group data consists of one item, the update interval. The update interval specifies the time period in milliseconds between DAServer reads of the device memory. You can specify a number between 0 and 2147483646 (596.52 hours).

To set the update interval

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Device Groups** tab.
- 4 Move the pointer over interval to be updated. Right-click the interval and click **Modify Update Interval**. The current update interval is selected.


Update Interval (ms)	
1000	

- 5 Type an interval and press Enter.
- 6  Click the **Save** icon.

Deleting a Device Group

When you delete a device group, the quality of items being accessed using the device group change to bad, and the DASServer rejects new requests for data using the device group.

To delete a device group

- 1 In the DASServer Manager, navigate to the device.
 - a Expand **DASServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DASServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Device Groups** tab.
- 4 Move the pointer over the device group to be deleted. Right-click the group and click **Delete**.
- 5 Read the warning and then click **Yes**.
- 6  Click the **Save** icon.

Chapter 5

Managing Device Items

Defining device items provides a more user-friendly way to name data in the device. Defining device items is optional. Use device items to access data in the DAServer and devices connected to the DAServer. Device items consist of two pieces: a name and an item reference. You can specify either from your client.

The device item name is an alternative name for the item reference. It is an “alias” or a label for the data in the device. You can use this label instead of the item reference when you create the client application.

The item reference identifies data in the device. The item reference is a PLC memory reference. Each device’s memory reference can have a different format. For more information, see *Item Reference Descriptions* on page 43.

The actual item reference can be entered as the device item name. In this case, the item reference value can be left empty.

To provide diagnostic and operational information, the DAServer has several system items that do not access data in a device. They are grouped by function:

- Global system items
- Device-group-specific system items
- Device-specific system items

For more information, see *Standard System Items* on page 43.

Device item names defined in the DAServer will show up as OPC browsable items.

You can add device items while the DAServer is active, and these new items are immediately available to client applications.

You can make changes to items while the DAServer is active. Changes take effect immediately. OPC clients that are already connected to the item are not affected until they release and re-acquire the item.

For detailed formats for specifying item references, see [Item Reference Descriptions](#) on page 43.

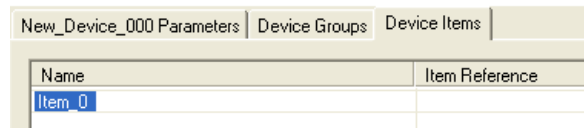
For information on how to subscribe to data items, see [Accessing the Data in Your DAServer](#) on page 41.

Creating Aliases for Item References

The device item name is an alias for the item reference. Device item names can be 256 characters long. Long names may be more explanatory, but your client application may have limited space for displaying names.

To set a device item name

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Device Items** tab.
- 4 Move the pointer over the columns, right-click and click **Add**. A **Name** field is selected with a default name.



- 5 Type the name and press **Enter**.
- 6 Click the **Save** icon.

Renaming a Device Item

Changing a device item name affects new client requests for data. Requests for data already accepted by the DAServer are not affected.

To change a device item name

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Device Items** tab.
- 4 Move the pointer over the name to be changed, right-click and click **Rename**.
- 5 Type the new name and press Enter.



- 6 Click the **Save** icon.

Deleting a Device Item

Deleting a device item name affects new client requests for data. Requests for data already accepted by the DAServer are not affected.

To delete a device item name

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Device Items** tab.
- 4 Move the pointer over the name to be deleted, right-click and click **Delete**.
- 5 Read the warning and click **Yes**.



- 6 Click the **Save** icon.

Setting the Item Reference

You must know which memory locations you need and the memory location attributes before entering item references in the DAServer.


For more information, see **Format and Syntax** on page 53.

For tables that list the options for each device type, see **Item Reference Descriptions** on page 43. In this case device type does not refer to the model of PLC.

To set an item reference

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Device Items** tab.
- 4 Move the pointer over the **Item Reference** field to be set, right-click, and click **Rename**.

New_Device_000 Parameters	
Device Groups	
Device Items	
Name	Item Reference
Item_0	R0004

- 5 Type the item reference and press Enter.
- 6  Click the **Save** icon.

Exporting and Importing CSV Files

To help you manage item references (tags) and device item names outside of the DAServer Manager, the DAServer supports the import and export of device item data in a comma separated value (CSV) file. The CSV functions are only available when a device items tab is selected.

To export a device item list

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Device Items** tab.
- 4 Move the pointer over the columns area, right-click, and click **Export**.
- 5 In the **Save As** dialog box, type a file name, select a directory, and press **Save**.

To import a device item list


- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.

- b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Device Items** tab.
- 4 Move the pointer over the columns area, right-click, and click **Import**.
- 5 In the **Open** dialog box, find the file containing the items to be imported, and press **Open**. Items contained in the file are now listed on the **Device Items** tab.

Clearing All Device Item Names

You can delete all device items for a device.

To clear all device item names

- 1 In the DAServer Manager, navigate to the device.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer, expand **Configuration**, and then expand the channel.
- 2 Select the device.
- 3 In the right pane, click the **Device Items** tab.
- 4 Move the pointer over the columns area, right-click, and then click **Clear All**.
- 5 Read the warning and click **Yes**.
-  6 Click the **Save** icon.

Chapter 6

Managing Your DAServer

After you configure the DAServer, there are two steps to take before you can access data with your client application.

The first step is to determine what kind of client applications are to be used with this DAServer. If any of your client applications use DDE/SuiteLink, you must configure the DAServer as a service. If only OPC client applications will be used, you can configure the DAServer as a service or as not a service.

The last step is to activate the DAServer. Some client applications can programatically activate the DAServer. If you configure the DAServer as an automatic service, the DAServer is started and activated when the computer on which the DAServer is installed starts up. If you configure the DAServer as a manual service, the DAServer is not started when the computer starts up. Instead, it is started upon the first connection from an OPC client or when activated from the DAServer Manager.

After a DAServer is running in auto or manual as a service, it stays running until explicitly stopped in the DAServer Manager or the computer shuts down.

Configuring the DAServer as Service

To support DDE/SuiteLink clients, the DAServer must be configured as a service.

To configure the DAServer as a service

- 1 In the DAServer Manager, navigate to the DAServer.
 - Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.

- 2 Right-click `Archestra.DASMTFXSerial.1` and then click **Configure As Service**.
- 3 Click either **Auto Service** or **Manual Service**.
- 4 Read the warning message and click **Yes**.

Configuring the DAServer as Not a Service

The DAServer can only be set to run as not a service when the DAServer is in the deactivated state.

To configure the DAServer as not a service

- 1 In the DAServer Manager, navigate to the DAServer.
 - Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
- 2 Right-click `Archestra.DASMTFXSerial.1` and then click **Configure As Service**.
- 3 Click **Not a Service**.
- 4 Read the warning message and click **Yes**.

Archiving Configuration Sets

A configuration set includes the DAServer's global parameters; each channel and its parameters; and each device and its parameters, device groups, and device items. It lets you manage the settings of different DAServer configurations.

The DAServer contains a default configuration set named `dasmtfxserial`. You cannot delete this default configuration set.

You can create multiple configuration sets and switch between them. Archiving, clearing, and switching configuration sets can only be done when the DAServer is deactivated.

Before you create a configuration set, verify that you have saved any changes you made to the global parameters. If you change a parameter and then immediately create a configuration set, the original parameter value is saved as part of the configuration set, not the changed value.



To save a global parameter, click the **Save** icon.

To archive a configuration set

- 1 In the DAServer Manager, navigate to the configuration node.

- a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
- b Expand the **DAServer**.
- 2 Click **Configuration**.
- 3 Right-click and click **Archive Configuration Set**.
- 4 In the dialog box, type the configuration set name, and click **Archive**. All the current configuration values are saved to the set.

After you archive at least one configuration set, you can select it for use.

To select a configuration set

- 1 In the **DAServer Manager**, navigate to the configuration node.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the **DAServer**.
- 2 Click **Configuration**.
- 3 Right-click, point to **Use Another Configuration Set**, then click the desired name.

To change the parameter values saved in a configuration set, make sure the desired configuration set is shown, then follow this procedure.

To change the parameter values in a configuration set

- 1 In the **DAServer Manager**, navigate to the configuration node.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the **DAServer**.
- 2 Click **Configuration**.
- 3 Change the parameters that you want to change.



- 4 Click the **Save** icon.

Clearing a configuration set returns the parameters to their default values.

To clear a configuration set

- 1 In the **DAServer Manager**, navigate to the configuration node.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the **DAServer**.

- 2 Click **Configuration**.
- 3 Right-click, move the mouse over **Clear Configuration Set**, then left click.
- 4 Read the warning message, then click **Yes**. The parameters are set to the default values.

To delete a configuration set

- 1 In the DAServer Manager, navigate to the configuration node.
 - a Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
 - b Expand the DAServer.
- 2 Click **Configuration**.
- 3 Right-click, move the mouse over **Delete Configuration Set**, then select the configuration set to delete.
- 4 Read the warning message, then click **Yes**.

Activating/Deactivating the DAServer

When you activate the DAServer, it starts communicating and accepting requests from client applications. If a DAServer is configured as an automatic service, the DAServer is started and activated when the computer starts up. Also, a DAServer can be activated by the an OPC client connection request.

To activate the DAServer

- 1 In the DAServer Manager, navigate to the DAServer.
 - Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
- 2 Right-click **Archestra.DASMTFXSerial.1** and then click **Activate Server**.

Deactivating your DAServer stops it from communicating with client applications.

A DAServer with active OPC clients does not stop until the last OPC client shuts down.

To deactivate the DAServer

- 1 In the DAServer Manager, navigate to the DAServer.
 - Expand **DAServer Manager**, expand the node group, and then expand **Local** or the remote computer name.
- 2 Right-click **Archestra.DASMTFXSerial.1** and then click **Deactivate Server**.
- 3 Read the warning message and click **Yes**.

In-Proc/Out-of-Proc

The DAServer can run as a stand-alone process (out-of-proc) or as part of the client process (in-proc).

When the DAServer is running out-of-proc, it supports requests from both DDE/SuiteLink and OPC client applications. When the DAServer is running in-proc, it only supports OPC client applications.

If the DAServer is running as a service, the icon on the DAServer node in the SMC is yellow. If the DAServer is running as not a service, the icon is white. For more information, see the DAServer Manager User's Guide.

Hot Configuration

The DAServer allows certain configuration parameters to be changed while the DAServer is active. See the sections about the specific parameters for limitations or constraints.

Demo Mode

You can install the DAServer without a license. The DAServer runs without a license in Demo mode for 120 minutes. While in demo mode the DAServer checks for a license every 30 seconds. When the 120 minutes expires:

- The DAServer stops updating items.
- All non-system items have a Bad quality status.
- New items are rejected.

After the 120 minutes the DAServer checks for a license every thirty seconds. If a license is not found, the DAServer logs a warning.

You can use the `$$SYS$Licensed` system item to check the status of your license. This item returns true if the proper license is found or the DAServer is in demo mode (the 120 minutes), otherwise, it returns false.

After the DAServer finds a valid license, it logs a message, stops looking for a license, and begins running normally. For more information, see the License Utility User Guide.

Chapter 7

Accessing the Data in Your DAServer

Client applications read and write to data items that are internal to the DAServer, as well as to the items located in the devices. Client application communication with the DAServer is done using either the OPC, or DDE/SuiteLink protocols. The client application may or may not be on the same computer as the DAServer.

You do not need to create device items in the DAServer for your OPC client applications.

For information on how to specify item references, see [Format and Syntax](#) on page 53. For information on specific address formats, see [Address Descriptions](#) on page 54.

Accessing Data Using OPC

To connect to the DAServer with an OPC client application, be aware of the following six parameters:

- **Node Name:** The computer name identifying the node where the DAServer is located. Only required for remote access.
- **Program Name:** ArchestrA.DASMTFXSerial.1
- **Group Name:** An OPC group defined and created by the client. The DAServer device group is used as the OPC access path.
- **Device Group:** A device group as defined on the DAServer. If omitted, the default device group is assumed.
- **Link Name:** The hierarchy of nodes names, from the channel node to the device node, separated by delimiters.
- **Item Name:** The specific data element. This can be the device item name or the item reference.

The combination of the link name and item name form the OPC data path for any OPC client to access DAServer data.

If the item specified is not valid for the device location, the DAServer does not accept the item, returns a bad quality, and generates an error message in the logger.

Accessing Data Using DDE/SuiteLink

The DDE/SuiteLink address has four fields:

- **Node Name:** The computer name identifying the node where the DAServer is located. Only required for remote access.
- **Application Name:** DASMTFXSerial
- **Topic Name:** A device group defined for the device.
- **Item Name:** The specific data element. This can be the device item name or the item reference.

The DDE/SuiteLink topic is the equivalent to the device group.

Chapter 8

Item Reference Descriptions

Item references access system items and read and write data from devices.

Standard System Items

System items supply you easy access to DAServer status and diagnostics information. Client applications can read data from them just like ordinary items. However, in most cases the system item values are not directly acquired through the communications layer. System item values are usually generated through internal calculations, measurements, and tracking by the DAS Engine.

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): A collection of items on the same physical location with the same protocol update rate.

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. `$$SYS$Status` at the root level of the whole hierarchy tree is always good, as it represents the quality status of the local computer itself. For practical application, OPC clients should reference `$$SYS$Status` at any hierarchy levels other than the root.

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.

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
<code>\$\$SYS\$Licensed</code>	Boolean/Read	Binary status indication of the existence of a valid license for the <code>DAServer</code> . If <code>FALSE</code> , this item causes the <code>DAServer</code> to stop updating existing tags, to refuse activation of new tags, and to reject write requests in addition to setting quality for all items to <code>BAD</code> . If <code>TRUE</code> , the <code>DAServer</code> functions as configured. All instances have the same value.	RANGE: 0, 1 1: Valid license exists. 0: No valid license exists.

Device-Group-Specific System Items

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

System Item Name (Type)	Type/Access Rights	Description	Values
\$SYS\$updateInterval	DWord/Read Write	Accesses 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 non-system items on that topic are not 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	Not supported by this DAServer.	Always returns zero.

System Item Name (Type)	Type/Access Rights	Description	Values
\$SYS\$WriteComplete	Integer/Read Write	<p>Accesses 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.</p> <p>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.</p> <p>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.</p> <p>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).</p> <p>If the value of this item is zero, it cannot be poked.</p>	<p>RANGE: -1, 0, 1</p> <p>1: Write complete (no writes are pending – initial state).</p> <p>0: Writes are pending.</p> <p>-1: Writes completed with errors.</p>

System Item Name (Type)	Type/Access Rights	Description	Values
\$SYS\$ReadComplete	Integer/ ReadWrite	<p>Accesses 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.</p> <p>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.</p> <p>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.</p>	<p>RANGE: -1, 0, 1</p> <p>1: Read complete (all values have been read).</p> <p>0: Not all values have been read.</p> <p>-1: All values have been read but some have a non-good quality.</p>
\$SYS\$ItemCount	DWord/Read	<p>Accesses the number of items in the corresponding device group. This item is read-only.</p>	<p>RANGE: 0...2147483647</p> <p>>=0: Number of items.</p>
\$SYS\$ActiveItemCount	DWord/Read	<p>Accesses the number of active items in the corresponding device group. This item is read-only.</p>	<p>RANGE: 0...2147483647</p> <p>>=0: Number of active items.</p>

System Item Name (Type)	Type/Access Rights	Description	Values
\$SYS\$ErrorCount	DWord/Read	Accesses 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.
\$SYS\$PollNow	Boolean/Read Write	Not supported by this 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>Binary status indication of the connection state to the device (hierarchy level) the item is attached to. The device group (OPC access path/topic) does not affect the value.</p> <p>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. \$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>

System Item Name (Type)	Type/Access Rights	Description	Values
\$SYS\$ErrorCode	Longint/Read	Detailed error code of the communications state to the device. The device group (OPC access path/topic) does not affect the value.	>=0: Good status (0 is the default state – connected. >0: is some device state like: connecting, initializing, etc. <0: Error status (value indicates the error).
\$SYS\$ErrorText	String/Read	Detailed error string of the communications state of the device. The device group (OPC access path/topic) does not affect the value.	Descriptive text for the communications state corresponding to the error code.

System Item Name (Type)	Type/Access Rights	Description	Values
\$SYS\$StoreSettings	Integer/Read Write	<p>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.</p> <p>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.</p> <p>If the update interval changes 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.</p> <p>Reading the item always provides 0. Read/Write values are persisted only if you set this system item. The values other than this persist only for the life of the DAServer.</p>	<p>RANGE: -1, 0, 1</p> <p>-1: Error occurred during saving the configuration file.</p> <p>0: Read value always if status is OK.</p> <p>1: Persist settings (cleared immediately).</p>

Supported Data Types

The data type is specified as a suffix in the item syntax. The DAServer supports the following data types.

Data Type	Description
Boolean	Single bit
Word	Unsigned 16 bit value. Bit 0 is the low bit. Bit 15 is the high bit.
Short	Signed 16 bit value. Bit 0 is the low bit. Bit 14 is the high bit. Bit 15 is the sign bit
DWord	Unsigned 32 bit value. Bit 0 is the low bit. Bit 31 is the high bit.
Long	Signed 32 bit value. Bit 0 is the low bit. Bit 30 is the high bit. Bit 31 is the sign bit.
Float	32 bit floating point value. A floating point value is two consecutive registers. The first registers the low word and the second registers the high word.
String	Null terminated ASCII string support. Includes HiLo LoHi byte order selection and string lengths up to 64 bytes.

Format and Syntax

Item references have four attributes:

- **Device Type** defines what the memory location can be used for. Device types include inputs, outputs, link relays, latch relays, and others. For device type limitations, see your PLC documentation. In this case, device type does not refer to the model of PLC.
- **Range** defines the valid addresses. Addresses are usually specified as one or more letters indicating the device type and a number. The number is specified as a hex or decimal value from 0 to an upper limit. Hexadecimal numbers must always start with a valid decimal digit. For example, the offset C3 should be written as 0C3.
- **Data Type** defines which data type the client request can specify when accessing the memory location. Examples are Short and Word only. For a complete list of valid data types, see Supported Data Types on page 52. The data type is often optional because most memory locations have a default data type.
- **Access** defines what you may do. Some memory locations are read only or write only, others are read/write.

You can access addresses using the optional data types by appending the data type after the address in the form: address@datatype. For example, R0@Long or R0 L.

For tables showing allowable usage for each memory address, see Address Descriptions on page 54.

String Access

The string data type uses a number appended to the address to specify the length of the string in bytes. An example is DSH020.4, which specifies four bytes starting at address 20. The valid length of a string in D memory is 2 to 64 bytes. The string length must be an even number.

The string length, appended to the address, is always in decimal notation.

The last number that you can access by a string is one less than the maximum available address range in the PLC. For example, for FX1S, D000 to D255 are valid registers in the PLC. So,

DSH000.64 accesses registers D0000 to D0031.

DSH223.64 accesses registers D0223 to D0254.

DSH254.02 accesses register D254.

DSH255.02 is invalid, as it attempts to access registers beyond D0254.

DSH250.64 is invalid as it also attempts to access registers beyond D0254.

See the Address Descriptions on page 54.

Overriding the Update Interval or Data Type

You can override the update interval on a per-tag basis by appending @[Update Interval] to the address. For example,

- To override just the update interval: R0001@500
- To override both update interval and data type: R0001@500, short.

Address Descriptions

The address descriptions consist of the device type, its item name; and the allowable range of values, the default data type, allowable suffix values, and allowable access methods.

Some range values are specified as hexadecimal numbers. Hexadecimal numbers must always start with a valid decimal digit. For example, the offset C3 should be written as 0C3.

Model FX Address Descriptions

Model FX supports the following types of devices.

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
Inputs	Xn	n=000 to 377 (Octal)	Boolean		Read Only
Outputs	Yn	n=000 to 377 (Octal)	Boolean		Read/Write
Internal Relays	Mn	n=0000 to 1535	Boolean		Read/Write

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
Special Internal Relays	Mn	n=8000 to 8255	Boolean		Read/Write
States	Sn	n=000 to 999	Boolean		Read/Write
Timer Contacts	TSn	n=000 to 255	Boolean		Read Only
Counter Contacts	CSn	n=000 to 255	Boolean		Read Only
Timer Reset	TRn	n=000 to 255	Boolean		Read/Write
Counter Reset	CRn	n=000 to 255	Boolean		Read/Write
Timer Value	Tn	n=000 to 255	Word	Short	Read/Write
Counter Value	Cn	n=00 to 255	Word	Short	Read/Write
32 Bit Counter Value	Cn	n=200 to 255	DWord	Long	Read/Write
Data Registers	Dn	n=000 to 7999 n=000 to 7998	Short	Word Long, DWord, Float	Read/Write
Data Registers String Access HiLo Byte Ordering	DSHd.l	d = 0000 to 7998 l = 2 to 64* *String length must be an even number.	String		Read/Write
Data Registers String Access LoHi Byte Ordering	DSLd.l	d = 0000 to 7998 l = 2 to 64* *String length must be an even number.	String		Read/Write
Special Data Registers	Dn	n=8000 to 8255 n=8000 to 8254	Short	Word Long, Float	Read/Write

Model FX0 Address Descriptions

Model FX0 supports the following device types.

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
Inputs	Xn	n=000 to 017 (Octal)	Boolean		Read Only
Outputs	Yn	n=000 to 15 (Octal)	Boolean		Read/Write
Internal Relays	Mn	n=0000 to 0511	Boolean		Read/Write
Special Internal Relays	Mn	n=8000 to 8255	Boolean		Read/Write
States	Sn	n=00 to 63	Boolean		Read/Write
Timer Contacts	TSn	n=00 to 55	Boolean		Read Only
Counter Contacts	CSn	n=00 to 15 n=235 to 254	Boolean		Read Only
Timer Reset	TRn	n=00 to 55	Boolean		Read/Write
Counter Reset	CRn	n=00 to 15 n=235 to 254	Boolean		Read/Write
Timer Value	Tn	n=00 to 55	Word	Short	Read/Write
Counter Value	Cn	n=00 to 15	Word	Short	Read/Write
32 Bit Counter Value	Cn	n=235 to 255	DWord	Long	Read/Write
Data Registers	Dn	n=00 to 31 n=00 to 30	Short	Word Long, DWord, Float	Read/Write
Data Registers String Access HiLo Byte Ordering	DSHd.l	d = 00 to 30 l = 2 to 62* *String length must be an even number.	String		Read/Write

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
Data Registers String Access LoHi Byte Ordering	DSLd.l	d = 00 to 30 l = 2 to 62* *String length must be an even number.	String		Read/Write
Special Data Registers	Dn	n=8000 to 8069 n=8000 to 8068	Short	Word Long, DWord, Float	Read/Write

Model FX0N Address Descriptions

Model FX0N supports the following types of devices.

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
Inputs	Xn	n=000 to 377 (Octal)	Boolean		Read Only
Outputs	Yn	n=000 to 377 (Octal)	Boolean		Read/Write
Internal Relays	Mn	n=0000 to 0511	Boolean		Read/Write
Special Internal Relays	Mn	n=8000 to 8255	Boolean		Read/Write
States	Sn	n=000 to 127	Boolean		Read/Write
Timer Contacts	TSn	n=00 to 63	Boolean		Read Only
Counter Contacts	CSn	n=00 to 31 n=235 to 254	Boolean		Read Only
Timer Reset	TRn	n=00 to 63	Boolean		Read/Write
Counter Reset	CRn	n=00 to 31 n=235 to 254	Boolean		Read/Write
Timer Value	Tn	n=00 to 63	Word	Short	Read/Write
Counter Value	Cn	n=00 to 31	Word	Short	Read/Write

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
32 Bit Counter Value	Cn	n=235 to 255	DWord	Long	Read/Write
Data Registers	Dn	n=000 to 255 n=000 to 254	Short	Word Long, DWord, Float	Read/Write
Data Registers String Access HiLo Byte Ordering	DSHd.l	d = 000 to 254 l = 2 to 64* *String length must be an even number.	String		Read/Write
Data Registers String Access LoHi Byte Ordering	DSLd.l	d = 000 to 254 l = 2 to 64* *String length must be an even number.	String		Read/Write
Special Data Registers	Dn	n=8000 to 8255 n=8000 to 8254	Short	Word Long, DWord, Float	Read/Write

Model FX0S Address Descriptions

Model FX0S supports the following types of devices.

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
Inputs	Xn	n=00 to 17 (Octal)	Boolean		Read Only
Outputs	Yn	n=00 to 15 (Octal)	Boolean		Read/Write
Internal Relays	Mn	n=000 to 511	Boolean		Read/Write
Special Internal Relay	Mn	n=8000 to 8255	Boolean		Read/Write
States	Sn	n=00 to 63	Boolean		Read/Write
Timer Contacts	TSn	n=00 to 55	Boolean		Read Only

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
Counter Contacts	CSn	n =00 to 15 n=235 to 254	Boolean		Read Only
Timer Reset	TRn	n=00 to 55	Boolean		Read/Write
Counter Reset	CRn	n =00 to 15 n=235 to 254	Boolean		Read/Write
Timer Value	Tn	n=00 to 55	Word	Short	Read/Write
Counter Value	Cn	n=00 to 15	Word	Short	Read/Write
32 Bit Counter Value	Cn	n=235 to 255	DWord	Long	Read/Write
Data Registers	Dn	n=00 to 31 n=00 to 30	Short	Word Long, DWord Float	Read/Write
Data Registers String Access HiLo Byte Ordering	DSHd.l	d = 00 to 30 l = 2 to 62* *String length must be an even number.	String		Read/Write
Data Registers String Access LoHi Byte Ordering	DSLd.l	d = 00 to 30 l = 2 to 62* *String length must be an even number.	String		Read/Write
Special Data Registers	Dn	n=8000 to 8255 n=8000 to 8254	Short	Word Long, DWord, Float	Read/Write

Model FX1N Address Descriptions

Model FX1N supports the following types of devices.

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
Inputs	Xn	n=00 to 177 (Octal)	Boolean		Read Only
Outputs	Yn	n=00 to 177 (Octal)	Boolean		Read/Write

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
Internal Relays	Mn	n=000 to 511	Boolean		Read/Write
Special Internal Relay	Mn	n=512 to 1535	Boolean		Read/Write
States	Sn	n=000 to 999	Boolean		Read/Write
Timer Contacts	TSn	n=00 to 255	Boolean		Read Only
Counter Contacts	CSn	n=000 to 255	Boolean		Read Only
Timer Reset	TRn	n=00 to 255	Boolean		Read/Write
Counter Reset	CRn	n=000 to 255	Boolean		Read/Write
Timer Value	Tn	n=00 to 255	Word	Short	Read/Write
Counter Value	Cn	n=000 to 199	Word	Short	Read/Write
32 Bit Counter Value	Cn	n=200 to 255	DWord	Long	Read/Write
Data Registers	Dn	n=0000 to 7999 n=0000 to 7998	Short	Word Long, DWord, Float	Read/Write
Data Registers String Access HiLo Byte Ordering	DSHd.l	d = 0000 to 7998 l = 2 to 64* *String length must be an even number.	String		Read/Write
Data Registers String Access LoHi Byte Ordering	DSLd.l	d = 0000 to 7998 l = 2 to 64* *String length must be an even number.	String		Read/Write
Special Data Registers	Dn	n=8000 to 8255 n=8000 to 8254	Short	Word Long, DWord, Float	Read/Write

Model FX1S Address Descriptions

Model FX1S supports the following types of devices.

Device Type	Item	Name Range	Data Type	Supported Suffixes	Access
Inputs	Xn	n=00 to 17 (Octal)	Boolean		Read Only
Outputs	Yn	n=00 to 15 (Octal)	Boolean		Read/Write
Internal Relays	Mn	n=000 to 511	Boolean		Read/Write
Special Internal	Mn	n=8000 to 8255	Boolean		Read/Write
Relay States	Sn	n=000 to 127	Boolean		Read/Write
Timer Contacts	TSn	n=00 to 63	Boolean		Read Only
Counter Contacts	CSn	n =00 to 31 n=235 to 254	Boolean		Read Only
Timer Reset	TRn	n=00 to 63	Boolean		Read/Write
Counter Reset	CRn	n=00 to 31 n=235 to 254	Boolean		Read/Write
Timer Value	Tn	n=00 to 63	Word	Short	Read/Write
Counter Value	Cn	n=00 to 31	Word	Short	Read/Write
32 Bit Counter Value	Cn	n=235 to 255	DWord	Long	Read/Write
Data Registers	Dn	n=000 to 255 n=000 to 254	Short	Word Long, DWord, Float	Read/Write
Data Registers String Access HiLo Byte Ordering	DSHd.l	d = 000 to 254 l = 2 to 64* *String length must be an even number.	String		Read/Write
Data Registers String Access LoHi Byte Ordering	DSLd.l	d = 000 to 254 l = 2 to 64* *String length must be an even number.	String		Read/Write

Device Type	Item	Name Range	Data Type	Supported Suffixes	Access
Special Data Registers	Dn	n=8000 to 8255	Short	Word	Read/Write
		n=8000 to 8254		Long, DWord, Float	

Model FX2N Address Descriptions

Model FX2N supports the following types of devices.

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
Inputs	Xn	n=000 to 377 (Octal)	Boolean		Read Only
Outputs	Yn	n=000 to 377 (Octal)	Boolean		Read/Write
Internal Relays	Mn	n=0000 to 3071	Boolean		Read/Write
Special Internal Relays	Mn	n=8000 to 8255	Boolean		Read/Write
States	Sn	n=000 to 999	Boolean		Read/Write
Timer Contacts	TSn	n=000 to 255	Boolean		Read Only
Counter Contacts	CSn	n=000 to 255	Boolean		Read Only
Timer Reset	TRn	n=000 to 255	Boolean		Read/Write
Counter Reset	CRn	n=000 to 255	Boolean		Read/Write
Timer Value	Tn	n=000 to 255	Word	Short	Read/Write
Counter Value	Cn	n=000 to 199	Word	Short	Read/Write
32 Bit Counter Value	Cn	n=200 to 255	DWord	Long	Read/Write
Data Registers	Dn	n=000 to 7999	Short	Word	Read/Write
		n=000 to 7998		Long, DWord, Float	

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
Data Registers String Access HiLo Byte Ordering	DSHd.l	d = 0000 to 7998 l = 2 to 64* *String length must be an even number.	String		Read/Write
Data Registers String Access LoHi Byte Ordering	DSLd.l	d = 0000 to 7998 l = 2 to 64* *String length must be an even number.	String		Read/Write
Special Data Registers	Dn	n=8000 to 8255 n=8000 to 8254	Short	Word Long, DWord, Float	Read/Write

Model FX2NC Address Descriptions

Model FX2NC supports the following types of devices.

Device Type	Item Name	Range	Data Type	Supported Suffixes	Access
Inputs	Xn	n=00 to 377 (Octal)	Boolean		Read Only
Outputs	Yn	n=00 to 377 (Octal)	Boolean		Read/Write
Internal Relays	Mn	n=0000 to 3071	Boolean		Read/Write
Special Internal Relay	Mn	n=8000 to 8255	Boolean		Read/Write
States	Sn	n=000 to 999	Boolean		Read/Write
Timer Contacts	TSn	n=000 to 255	Boolean		Read Only
Counter ontacts	CSn	n=00 to 255	Boolean		Read Only
Timer Reset	TRn	n=000 to 255	Boolean		Read/Write
Counter Reset	CRn	n=00 to 255	Boolean		Read/Write
Timer Value	Tn	n=000 to 255	Word	Short	Read/Write
Counter Value	Cn	n=000 to 199	Word	Short	Read/Write
32 Bit Counter Value	Cn	n=200 to 255	DWord	Long	Read/Write

Device Type	Item	Name Range	Data Type	Supported Suffixes	Access
Data Registers	Dn	n=0000 to 7999 n=0000 to 7998	Short	Word Long, DWord, Float	Read/Write
Data Registers String Access HiLo Byte Ordering	DSHd.l	d = 0000 to 7998 l = 2 to 64* *String length must be an even number.	String		Read/Write
Data Registers String Access LoHi Byte Ordering	DSLd.l	d = 0000 to 7998 l = 2 to 64* *String length must be an even number.	String		Read/Write
Special Data Registers	Dn	n=8000 to 8255 n=8000 to 8254	Short	Word Long, DWord, Float	Read/Write

Chapter 9

Troubleshooting

You can troubleshoot problems with the DAServer using the:

- Windows Task Manager
- Windows Performance and Alerts (PerfMon) application, also called Performance Monitor
- DAServer Manager
- ArcestrA Log Flag Editor
- ArcestrA Log Viewer

Your client application may let you view error messages, monitor the status of requests, and allow you to request data on the status of the DAServer and connected devices. For more information, see your client application documentation.

Troubleshooting with Windows Tools

Windows has two tools that may be useful in troubleshooting performance problems.

You can find quick verification that the DAServer process is running by looking at the Windows Task Manager. It also provides information on the user, CPU and memory usage of the processes.

If you need more information, or need to gather data while not logged in, you can use the Performance and Alerts application. For more information, see the Microsoft Management Console (MMC) help files on the Performance application. The Performance application is one of the administrative tools found in the Windows Control Panel.

Troubleshooting with the DAServer Manager

The DAServer Manager has information that may be useful in troubleshooting problems. When the DAServer is active, a diagnostic node is present below the configuration node in the console tree of the System Management Console.

Each diagnostic leaf contains information about DAServer activity. For more information, see the DAServer Manager User Guide or Help files.

Finding Version Information

If you contact Technical Support, you may need to supply version information.

To determine the DAServer Manager version

- In the DAServer Manager, right-click **DAServer Manager**, and then click **About DAServer Manager**. An about box appears showing the version and build date of the DAServer Manager.

To determine version information for DAServer components

- In the DAServer Manager, select the DAServer name in the console tree. The version information for each DAServer component is shown in the details pane.

Using the Wonderware Log Viewer

Error messages are created by the DAServer and logged by the Logger. You can view these messages with the Log Viewer. The Log Viewer help files explain how to view messages and how to filter which messages are shown.

Log Flags are categories of messages. The Log Flag Editor User Guide contains an explanation of the categories. Using the Log Flag Editor, you can specify which log flags the DAServer creates.

To open the Log Flag Editor

- 1 In the System Management Console, expand **Log Viewer** and then expand the log viewer group.
- 2 Select **Local**.
- 3 On the **Action** menu, click **Log Flags**.

In general, look at error and warning messages to determine if a problem exists. To determine whether the DASServer is communicating with a device, you can enable the DASSend and DASReceive log flags. From these you can determine whether or not the device is responding.

Note Generating large numbers of diagnostic messages can impact DASServer performance. You should not run in production with any more flags than those set when the DASServer is installed. To troubleshoot you can turn on more flags, but there is a performance impact. For more information, see the Log Flag Editor User Guide.

Basic Log Flags

The basic log flags for all ArchestrA components are:

- **Error:** A fatal error, the program cannot continue. By default set on by logger.
- **Warning:** The error is recoverable. A client called with a bad parameter, or the result of some operation was incorrect, but the program can continue. By default set on by logger.
- **Start-Stop:** Each main component logs a message to this category as it starts and stops.
- **Info:** These are general diagnostic messages.
- **Ctor-Dtor:** C++ classes of interest log messages to this category as they are constructed and destructed.
- **Entry-Exit:** Functions of interest log messages to this category as they are called and return.
- **Thread Start-Stop:** All threads should log messages to this category as they start and stop.

DAServer Log Flags

Messages created for these log flags are for DAServer common components and contain information about internal DAServer activities.

- **DACmnProtFail:** Some failure occurred in the common components while sending a message, updating an item, or otherwise moving data. Typically, this represents some unexpected behavior in the server-specific DLL.
- **DACmnProtWarn:** Some problem occurred that interfered with sending messages, updating items, or otherwise moving data. Common examples are slow poll, value limiting during type conversion, and transaction timeout messages.
- **DACmnTrace:** Normal processing of client program requests and data movement to and from the server-specific DLL are traced on this log flag. Use this in conjunction with **DACmnVerbose** to get the most information.
- **DACmnVerbose:** Many log flags used by the DAS common components are modified occasionally by **DACmnVerbose**. When **DACmnVerbose** is set, the logging of messages on other log flags includes more information.
- **DACmnSend:** Operations within the DAS Engine DLL that revolve around sending messages to the server-specific DLL.
- **DACmnReceive:** Events surrounding messages that are returned to the DAS Engine by the server-specific DLL, including the blocking and unblocking of hierarchies.

DAServer-Device Interface Log Flags

Messages created for the following log flags are specific to an individual DAServer and contain information about communications between the DAServer and device.

- **DASProtFail:** An error in the protocol occurred; for example, device disconnected. The program can continue, and, in fact, this category is expected during normal operation of the program. Must be set on by the generic DAS code when the DAServer starts.
- **DASProtWarn:** Something unexpected occurred in the protocol; for example, a requested item with an otherwise valid item name is not supported by this device. Must be set on by the generic DAS code when the DAServer starts.
- **DASTrace:** General diagnostic messages of a protocol-specific nature; for example, the number of items in a message for a specific protocol, if this number could be useful for optimizing.
- **DASVerbose:** Modifies all other DAS logging flags. When on, other messages may become more verbose, at the developer's discretion.
- **DASSend:** Protocol messages sent to the device are logged to this category.
- **DASReceive:** Protocol messages received from the device are logged to this category.
- **DASStateCat1, DASStateCat2, DASStateCat3, DASStateCat4:** These are general categories for use by the server developer. As DeviceEngine-generated state machines are created by the DAServer, they can be told to log state machine messages to one of the following: DASStateCat1, DASStateCat2, DASStateCat3, or DASStateCat4. These messages indicate when a state is made the active state, when a state handler is run, when a state handler completes, and when a timeout occurs for a state machine.
- **DASStateMachine:** By default, DeviceEngine-generated state machines created by the DAServer will log to this category, unless specifically told to log to one of the DASStateCatN categories. In addition, general state machine messages are logged to this category. These messages indicate when a state machine is created and deleted.

Error Message Descriptions

These are the messages specific to the Wonderware DAServer for Mitsubishi FX Serial. When you have the specified log flag enabled, these messages are generated and logged

Message	Log Flag	Possible Cause	Solution
Missing address	Warning	A tag address has no length.	Reenter the address in the client application.
Device address '<address>' contains a syntax error.	Warning	A tag address contains one or more invalid characters.	Reenter the address in the client application.
Address '<address>' is out of range for the specified device or register.	Warning	A tag address references a location that is beyond the range of supported locations for the device.	Verify the address is correct. If not, reenter it in the client application
Device address '<address>' is not supported by model '<model name>'.	Warning	A tag address references a location that is valid for the communications protocol but not supported by the target device.	Verify that the address is correct. If not, reenter it in the client application. Verify that the selected model name for the device is correct.
Data Type '<type>' is not valid for device address '<address>'	Warning	A tag address is assigned an invalid data type.	Modify the requested data type in the client application.
Device address '<address>' is read-only.	Warning	A tag address has a requested access mode that is not compatible with what the device supports for that address.	Change the access mode in the client application.
COMn does not exist.	Error	The specified COM port is not present on the target computer.	Verify the proper COM port is selected.

Message	Log Flag	Possible Cause	Solution
Error opening COMn.	Error	The specified COM port could not be opened due to an internal hardware or software problem on the target computer.	Verify the COM port is functional and can be accessed by other Windows applications.
COMn is in use by another application.	Error	The serial port assigned to a device is being used by another application.	Verify that you have assigned the correct port to the channel.
Unable to set comm parameters on COMn	Error	The serial parameters for the specified COM port are not valid.	Verify the serial parameters and make any necessary changes.
Communications error on COMn [<error mask>]	Error	<ol style="list-style-type: none"> 1. The serial connection between the device and the host PC is bad. 2. The communications parameters for the serial connection are incorrect. 	<ol style="list-style-type: none"> 1. Verify the cabling between the PC and the PLC device. 2. Verify the specified communications parameters match those of the device.
Device '<device name>' is not responding.	Error	<ol style="list-style-type: none"> 1. The serial connection between the device and the host PC is broken. 2. The communications parameters for the serial connection are incorrect. 3. The named device may have been assigned an incorrect network ID. 	<ol style="list-style-type: none"> 1. Verify the cabling between the PC and the PLC device. 2. Verify the specified communications parameters match those of the device. 3. Verify the network ID given to the named device matches that of the actual device.
Unable to write to '<address>' on device '<device name>.'	Error	<ol style="list-style-type: none"> 1. The serial connection between the device and the host PC is broken. 2. The communications parameters for the serial connection are incorrect. 3. The named device may have been assigned an incorrect network ID. 	<ol style="list-style-type: none"> 1. Verify the cabling between the PC and the PLC device. 2. Verify the specified communications parameters match those of the device. 3. Verify the network ID given to the named device matches that of the actual device.

Index

Symbols

\$SYS\$Status 44

A

activate 38

Address Descriptions 54

C

channel

adding 15

deleting 19

renaming 16

communication timeouts

fail after 23

reply timeout 23

configuration sets

archiving 36

changing parameters 37

clearing 37

deleting 38

selecting 37

configure

as a service 35

not a service 36

D

DAServer Manager 9, 14

troubleshooting 66

DASReceive 69

DASSend 69

data collection

enabling 23

data types

definition 53

override 54

supported 52

DDE/SuiteLink

\$SYS\$STATUS 44

accessing data using 42

configure as a service 35

device groups 25

topic 42

deactivate 38

demo mode 39

device groups 25, 43

adding 25

deleting 28

renaming 26

setting data 27

update interval 27

device items 29

clearing all 33

definition 29

deleting 31

renaming 30

setting the name 30

devices
 adding 21
 deleting 24
 enabling data collection 23
 renaming 22
 setting parameters 22
 setting the model 22
duty cycle 19

E

error descriptions 70
error log flag 67
export 32

F

fail after 23
firewall 11
flow control 17
FX Address Descriptions 54
FX0 Address Descriptions 56
FX0N Address Descriptions 57
FX0S Address Descriptions 58
FX1N Address Descriptions 59
FX1S Address Descriptions 61
FX2N Address Descriptions 62
FX2NC Address Descriptions 63

I

ID
 channel parameter 17
importing 32
in-proc 39
item reference 29
 access 53
 data type 53
 device type 53
 format and syntax 53
 memory access 53
 range 53
 setting 31
 string access 53
 supported data types 52

L

Log Flag Editor 67

log flags 66
 DASReceive 69
 DASSend 69
 error 67
 warning 67
Log Viewer 10

O

OPC
 \$SYS\$STATUS 44
 accessing data using 42
 browsable items 29
 device groups 25
 supported protocols 10
out-of-proc 39

R

reply timeout 23

S

service
 configuring as 35
 configuring not as a 36
SuiteLink
 address fields 42
 supported protocols 10
system items 10, 29, 43

T

time out 23
troubleshooting 10, 65
 Log Flag Editor 66
 Log Viewer 66
 version information 66

U

update interval 25, 27
 override 54

V

version information 66

W

warning log flag 67