Wonderware®

SIDirect DAServer User's Guide

Invensys Systems, Inc.

Revision 1.5

Last Revision: 12/19/06

Copyright

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

http://www.wonderware.com

Trademarks

Contents

Welcome

This guide describes using and customizing the SIDirect DAServer application.

You can view this document online or you can print it, in part or whole, by using the print feature in Adobe Acrobat Reader.

This guide assumes you know how to use Microsoft Windows, including navigating menus, moving from application to application, and moving objects on the screen. If you need help with these tasks, see the Microsoft online Help.

In some areas of the SIDirect DAServer, you can also right-click to open a menu. The items listed on this menu change, depending on where you are in the product. All items listed on this menu are available as items on the main menus.

About this Book

This user's guide describes configuring and using the Wonderware® SIDirect DAServer after you install it.

This book is organized in the following fashion:

- **Contents**
- **Introduction**
  Contains overview information about the DAServer, its features, and the environment in which it works.
- **Configuration**
  Contains a detailed description of the user-interface elements of this DAServer as well as its functionality.
- **Item Naming**
  Describes the item-naming conventions for targeted devices.

- **Troubleshooting**
  Provides information about error and warning messages displayed by this DAServer, monitoring connectivity status with the device or the status of DAS conversations, and debugging communications between the DAServer and the targeted device.
- **Reference**:
  Describes the DAServer architecture in general.
- **Index**

Documentation Conventions

This documentation uses the following conventions:

Convention	Used for
Initial Capitals	Paths and filenames.
Bold	Menus, commands, buttons, icons, 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

Introduction

This section is an overview of the Wonderware® SIDirect DAServer (Data Access Server), including the application-level and bus-level communications protocols, item naming conventions, and server features.

Overview

The SIDirect DAServer is a Microsoft® Windows® application program that acts as a communications protocol server. It provides other Windows application programs with access to data within the Siemens S7-200/300/400 family of PLCs. The SIDirect DAServer provides access to a Siemens PLC through an off-the-shelf standard Ethernet network interface card in the computer.

Note The DAServer can support multiple network interface cards in a system.

The SIDirect DAServer supports the Siemens S7-200/300/400 family of hardware and firmware listed in Supported DAServer Hardware and Firmware on page 151. It does **not** support the following items:

- Siemens PC adapter cards
- Siemens CP (Communications Processor) cards that reside in the PC
- Siemens SIMATIC NET library

While the SIDirect DAServer is primarily intended for use with the Wonderware InTouch® software (Version 7.11 Patch 02, 8.0, 9.0, or 9.5), it can be used by any Microsoft Windows program capable of acting as a DDE, FastDDE, SuiteLink™, or OPC client.

Communications Protocols

The SIDirect DAServer 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 TCP bus protocols over the Ethernet to communicate with the PLCs.

Note This DAServer is compliant with the OPC Data Access (DA) 2.05a specifications.

For more information about the DAServer architecture, see Reference on page 147.

Application Communications Protocols

This section describes the application communications protocols that can communicate with the clients.

Note SuiteLink, DDE, and OPC clients coexist with FactorySuite A^2.

OPC

OPC (OLE for Process Control) is a non-proprietary set of standard interfaces based upon the OLE/COM technology developed by Microsoft. This standard enables 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 writing custom drivers to be done one time, and then to be easily reused by HMI, SCADA, control, and custom applications.

Over the network, OPC uses DCOM (Distributed COM) for remote communications.

SuiteLink

SuiteLink uses a TCP/IP-based protocol 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 including Windows 2000 Professional, Windows 2000 Server, Windows 2000 Advanced Server, Windows 2003 Server, and Windows XP Pro.

SuiteLink is not a replacement for DDE orFastDDE. 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 timestamp 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 if the applications are on a single node or distributed over a large node count.
- The network transport protocol is TCP/IP using the Microsoft standard WinSock interface.

FastDDE

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

DDE

DDE is a communications protocol developed by Microsoft to allow applications in the Windows environment to send and receive data and instructions to and 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 or Microsoft Excel, can simultaneously be both a client and a server.

Bus Communications Protocols

The SIDirect DAServer uses only the TCP bus communications protocols over the Ethernet to communicate with the Siemens S7-200/300/400 family of controllers. The SIDirect DAServer does **not** support MPI, Profibus, and other non-Ethernet protocols.

Accessing Items via the DAServer

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

Accessing Items Using the OPC Communications Protocol

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 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 connects to the DAServer, either in-process or out-of-process, and creates 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. Private OPC groups are local to a single client.

Note DAServers only support private OPC groups.

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. This node name is required for remote access and is optional for local access.

- **program name**
  The registered OPC server name uniquely identifying a specific server (ProgID).

  For this DAServer, the program name is **ArchestrA.DASSIDirect.1**.

- **group name**
  The OPC group created from the client for logically organizing a collection of items 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 separated by delimiters. Link names represent 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.

- **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.

Connecting to the DAServer from OPC Client

If you connect to the DAServer using either a VB OPC client or a C OPC client with the option **All** selected, the DAServer is loaded in-process. This in-process connection prevents a SuiteLink client from connecting to the DAServer after the VB client is already connected. To enable a SuiteLink client to connect to the DAServer when a VB client is already connected, the DAServer needs to start in an out-of-process mode when the VB client first connects to it.

To fix this in-process connection problem, make sure that the OPCDAAuto.dll installed in your computer is the Wonderware OPCDAAuto.dll. If your computer also has another Wonderware OPCDAAuto.dll installed (for example a .dll file distributed by the OPC Foundation or other organizations), the change made in the Wonderware OPCDAAuto.dll to specifically fix the problem becomes ineffective and the in-process connection problem cannot be solved.

To find and use the Wonderware OPCDAAuto.dll File

1. Perform a global search on your computer to find the number of copies of OPCDAAuto.dll files installed.
2. Verify the version used is from Wonderware by right-clicking on the **OPCDAAuto.dll** file and selecting the **Properties** option and the **Version** tab.
   The normal or default installation directory of the OPCDAAuto.dll is C:\Program Files\Common Files\ArchestrA™.
3. Register the Wonderware OPCDAAuto.dll file by running: regsvr32 C:\Program Files\Common Files\ArchestrA\OPCDAAuto.dll.

Accessing Items Using the DDE/SuiteLink Communications Protocol

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

The following briefly describes each portion of this naming convention.

- **node name**
  Computer name or host name identifying a specific node on the network. This node name is required for remote access and is optional for local access.

- **application name**
  The name of the Windows program (this DAServer) that accesses the data element. In the case of data coming from or going to the Siemens devices via the DDE/SuiteLink Plugin of this DAServer, the application name portion of the address is **DASSIDirect**.

- **topic name**
  Meaningful names configured in the DAServer to identify specific devices. These names are used as the topic names in all conversations with that device.
  For example, **FASTPOLL**.
  Topic names map 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 or items 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.

For more information on item names, see Item Naming on page 43.

Features

The SIDirect DAServer provides the following features:

- Communicates over multiple application-level protocols at the same time
- Adds new application-level protocols on the fly
- Remote configuration
- New, robust diagnostic abilities
- Additional server-specific diagnostics
- XML storage
- Full, existing S7 support such as alarms and events, block services, cyclic services, regular S7 messages
- Full, existing item name space
- Log of errors, warnings, traces, and SAPI messages, individually adjustable for reading and writing
- OPC browsing
- Operates without the installation of the Siemens SIMATIC NET middleware
- Support of pass-through block read/write to S7 data blocks
- The use of timestamp in the alarm message when timestamping any alarm message data received from the PLC
- Support of specific Memory syntax (V, AI, AQ, SM, S, HC) for the S7-200 PLCs
- New Q syntax for outputs and PQ syntax for peripheral outputs
- Can ping the PLC connectivity even when there are no items subscribed to

For more in-depth information about the DAServer architecture, see Reference on page 147.

Demo Mode

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

When you first start this SIDirect 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 SIDirect DAServer repeats its request for the license every 30 seconds. If no licenses are 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 receive Bad quality status. The SIDirect DAServer continues to request a license and clients continue to function normally. For example, you can still add or remove an item, but its quality is set to Bad until a license is obtained.

Note Use the SYSLicensed 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 After a DAServer obtains a valid license, it no longer checks for a license. If your license expires while the DAServer is running, the DAServer continues to function normally until it is stopped.

Chapter 2

Configuration

After you install the Wonderware SIDirect DAServer, you need to configure it. Perform this configuration using the DAServer Manager hosted in the **ArchestrA System Management Console**. You can start the SMC through the **Programs** menu of the Windows **Start** button.

Before you activate the DAServer, you must first build the device hierarchy, simulating the physical hardware layout, to establish communications to each of the controllers. After you build the S7 hierarchy, you can configure the respective devices for communications.

Finally, you can create the desired Device Groups for each controller by clicking on the **Device Groups** tab.

Getting Started Quickly with the SIDirect DAServer

This section briefly describes the procedures required to prepare the SIDirect DAServer for use and is intended for people who are familiar with DAServers. If you are not familiar with the DAServer functionality, go to the detailed procedures in Configuring the SIDirect DAServer on page 21.

To prepare the SIDirect DAServer

1. Install the Wonderware SIDirect DAServer from Windows by running the **Setup.exe** program and accepting all the default settings during installation. If you need more information, the DAServer installation instructions are included in a separate Help file named Install-SIDirect.chm.

Important Because there are no default values for security settings, note the User Name and password selected during the install.

2 Start the Wonderware DAServer Manager. Click **Start**, point to **Programs**, then click **Wonderware** and **System Management Console**.

3 Click **ArchestrA System Management Console** and find the SIDirect DAServer in the **DAServer Manager** tree.

Under the local node the DAServer name is **ArchestrA.DASSIDirect.1**.

Note See the DAServer Manager Online Help for information about working in this snap-in environment.

4 Determine the hierarchical structure of the network/PLC environment to which you plan to connect.

5 Configure the new SIDirect DAServer. Select and right-click on the **Configuration** branch of the hierarchy and select **Add PortCpS7 Object**. The SIDirect DAServer allows only 1 (one) instance of PortCpS7 object in the hierarchy.

In this step, the hierarchy entry is added in edit mode. You can appropriately describe components of your specific hardware environment. If you do not rename the object at this time, a numeric sequencing system is applied. Any hierarchy entry can be renamed at a later time.

6 Select and right-click on the **New_PortCPS7_000** object you created in the tree. Select **Add S7Cp Object** to create an S7Cp object or **Add S7Cp_200 Object** to create an S7Cp_200 object.

The SIDirect DAServer allows up to 1024 S7Cp objects and 1024 S7Cp_200 objects in the hierarchy.

7 Optionally, create the desired device groups in the **Device Groups** dialog box of each logical end-point object.

Important 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 server resumes after the first one is closed.

Your SIDirect DAServer is now ready to use. To use the DAServer, you must activate it.

- If you use an OPC Client, the SIDirect DAServer auto-starts.
- If you use DDE/SuiteLink, start the SIDirect DAServer either as a manual or automatic service.
- To activate the DAServer, right-click on **ArchestrA.DASSIDirect.1** and select **Activate Server**.

To run the SIDirect DAServer as a service, right-click on the SIDirect **DAServer name** and select **Configure As Service**. You can configure it as an auto service or manual service.

For more information about configuring your SIDirect DAServer as a service, see the Activation/Deactivation/Service Component of the DAServer Manager documentation.

Configuring the SIDirect DAServer

The SIDirect DAServer is hosted by the DAServer Manager, a Microsoft Management Console (MMC) snap-in environment. Many high-level functions and user-interface elements of the DAServer Manager are universal to all DAServers. **Only** the documentation for the DAServer Manager contains descriptions of those universal functions/UI elements.

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, click the Help topics on the MMC **Help** menu. Both the MMC and DAServer Manager Help are opened. An Adobe Acrobat version of the DAServer Manager documentation (DAServerManager.pdf) is provided.

Note Most items in each shortcut menu are standard Windows commands. See the **Help** menu of the MMC for more information about those commands.

Before you can configure the SIDirect DAServer, you need to perform the following steps.

1. Install the Wonderware SIDirect DAServer by running the **Setup.exe** program and accept all the default settings during installation. If you need more information, the DAServer installation instructions are included in a separate Help file (Install-SIDirect.chm).

Important Because there are no default values for security settings, note the User Name and password selected during the install.

2 Run the DAServer Manager on at least one computer.

After you install the SIDirect DAServer, you need to configure it.

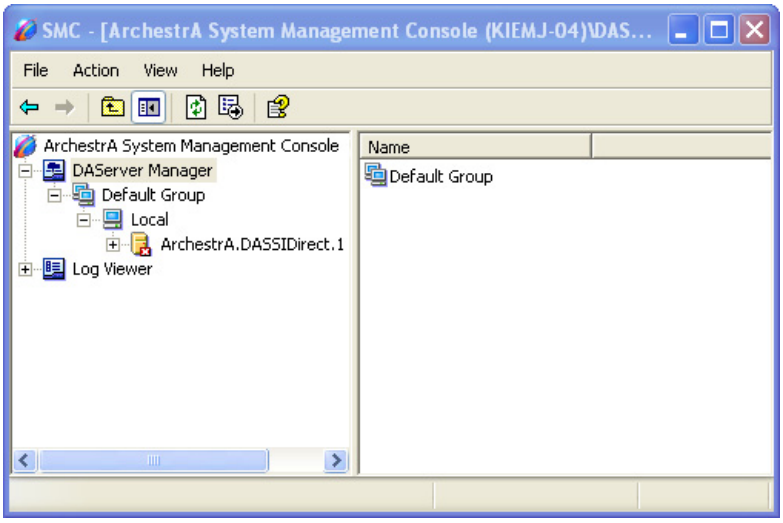
To prepare the SIDirect DAServer

1 Start the System Management Console by clicking the **Start** button on the Windows task bar and pointing to **Programs**. Point to the **Wonderware** folder that contains the System Management Console, click **ArchestrA System Management Console**, and click on **DAServer Manager**.

2 Click on **Default Group**, then click on **Local**.

Under the Local node, the name of the DAServer is **ArchestrA.DASSIDirect.1**.

Note See the DAServer Manager documentation for general information about working in this snap-in environment.



3 Before starting the DAServer, build the device hierarchy to establish communications to each of the controllers.

For step-by-step procedures on how to build the device hierarchy, see SIDirect Hierarchy in the DAServer Manager on page 23.

Important Selecting the **Configuration** object of the hierarchy tree opens the **Global Parameters** configuration view for this SIDirect DAServer. Configure all other global parameters as required for this SIDirect DAServer.
For more information about the **Global Parameters** dialog box, including descriptions of the different Poke Modes, see the DAServer Manager documentation.
Any Global Parameters that appear dimmed are not available.

4 After you build the SIDirect hierarchy, you can start configuring the respective devices for communications.

5 Finally, create the Device Groups for each controller. Navigate to the object in the **DAServer Manager** tree view and click on the **Device Groups** tab.

For step-by-step procedures on configuring Device Groups, see Configuring Device Group and Device Item Definitions on page 30.

Note If a 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 server resumes after the first one is closed.

The DAServer is ready to use after you activate it.

- If you use an OPC Client, the DAServer auto-starts.
- If you use DDE/SuiteLink, start the DAServer either as a manual or automatic service.
- To activate the DAServer, right-click on **ArchestrA.DASSIDirect.1** and select **Activate Server**.

To run the SIDirect DAServer as a service, right-click on the SIDirect **DAServer name** and select **Configure As Service**. You can configure it as an auto service or manual service.

For more information about configuring your SIDirect DAServer as a service, see the Activation/Deactivation/Service Component of the DAServer Manager documentation.

SIDirect Hierarchy in the DAServer Manager

The SIDirect DAServer uses a two-tier hierarchy for modeling the S7 objects and a custom device group configuration.

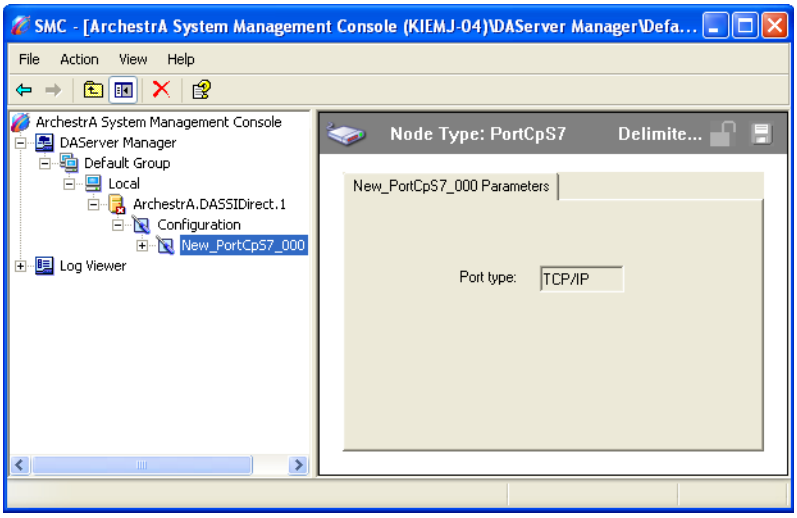
Before configuring your SIDirect DAServer, determine the hierarchical structure of your network/PLC environment.

PortCpS7 Object

The server-specific configuration portion of the SIDirect DAServer hierarchy tree under the DAServer Manager starts at the PortCpS7 (Communications Processor) object. It represents the network board in the computer that communicates with the PLC. Usually it is an ordinary network card identified by the local IP address.

To add PortCpS7 object to your SIDirect hierarchy

1 Select and right-click on the **Configuration** branch of the hierarchy, then select **Add PortCpS7 Object**. The **New_PortCpS7_000 Parameters** object appears.



The hierarchy entry is added in edit mode. You can appropriately describe components of your specific hardware environment. If you do not rename the object at this time, a numeric sequencing system is applied. Any hierarchy entry can be renamed at a later time

- Only 1 (one) PortCpS7 object can be created from the **Configuration** branch.

2 Rename this object as needed.

The New_PortCpS7_000 Parameters configuration view shows one element already configured:

- **Port type**
  The type of the port is TCP/IP.

Important If you subsequently clear your configuration hierarchy, you must create this PortCpS7 object by right-clicking on **Configuration** and selecting **Add PortCpS7 Object**. An object called New_PortCpS7_000 Parameters is created. Rename as appropriate. From this point, all of the following instructions apply.

S7Cp Object

The SIDirect DAServer can connect to different CPs in an S7 PLC. Each of the S7Cp node models the end-point of the communications path. The S7Cp object can access only some of the memory type of the S7-200 PLC because the S7-200 PLC has its own specific syntax.

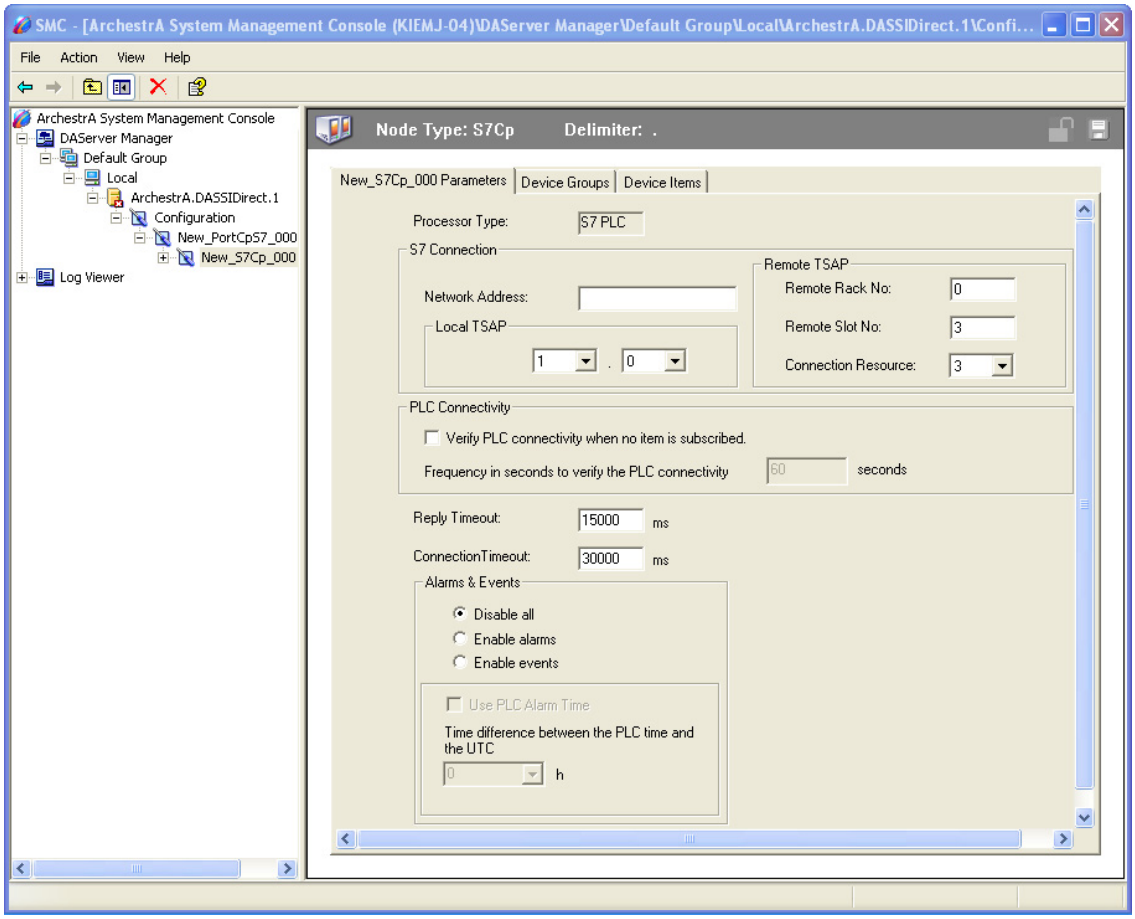
For the S7Cp object, you must pre-configure the connection information on the S7 200 PLC. The S7Cp_200 object can connect to the S7 200 PLC using the default settings.

From the New_PortCpS7_000 branch of the DAServer hierarchy, create the S7Cp object.

To add S7Cp object to your SIDirect hierarchy

1 Select and right-click on your

2 Select and right-click your **New_PortCpS7_000** object, then select **Add S7Cp Object**. The New_S7Cp_000 Parameters configuration view appears.



A maximum of 1024 S7Cp objects can be created from the **New_PortCpS7** branch.

3 Rename as needed.

The S7Cp configuration view has six elements, five of which you can configure:

- **Processor Type**
  The processor type is S7 PLC.
- **S7 Connection**
  The S7 Connection has the following three configurable parameters.
  - **Network Address**
    The IP address or host name of the remote S7 CP. Type in the network address where the PLC is located (for example, "10.11.12.13"), or a host name if one is defined in the local hostlist. The address cannot be more than 255 characters. The first and last character must be alphanumeric. The rest of the characters must be either alphanumeric, "-" (dash), or "." (period).
  - **Local TSAP**
    The local TSAP of the computer. Select the Hex numbers for the connection resources from the menu. The Local TSAP consists of two (2) Hex numbers. The first number corresponds to the connection resource. Each number ranges from 00 to FF. The default values are 01 and 00, respectively.
  - **Remote TSAP**
    This Remote TSAP corresponds to what you configured in the TSAP for the S7 CP. Configure the Remote TSAP by typing in the decimal numbers for the **Remote Rack** and **Remote Slot**, and by selecting the Hex number for the **Connection Resource** from the menu. The values for the **Remote Rack No.** and **Remote Slot No.** range from 0 to 255, with the default values of 0 and 3, respectively. The value for the **Connection Resource** ranges from 00 to FF. The default value is 03.
- **PLC Connectivity**

  The watchdog scheme for detecting the connectivity status to the PLC when there are no activities (no items are subscribed to).
  - Select the **Verify connectivity when no item is subscribed** check box to turn on the watchdog.
  - Specify the watchdog time interval, in seconds, in the **Frequency in seconds to verify the PLC connectivity** box.

- **Reply Timeout**
  Enter a value, in milliseconds, beyond which messages time out.
  - Allowable range is 0 to 100,000 milliseconds.
  - The default value is 15,000 milliseconds.

  If you decrease this value, the SIDirect DAServer reacts faster to a communications failure.
- **Connection Timeout**
  Enter a value, in milliseconds, beyond which a pending request to initiate a connection times out.
  - Allowable range is 0 to 100,000 milliseconds.
  - The default value is 30,000 milliseconds.
- **Alarms and Events**
  Enable Alarms or Events or disable both for this connection by selecting:
  - Disable all
  - Enable alarms
  - Enable events

  On one connection, either Alarms, Events, or none can be configured. If you need to access both Alarms and Events, create two different connections.

  Use PLC Alarm Time
  The alarm time in the PLC used to timestamp alarm and event-related data.

  Time Difference Between the PLC Time and the UTC
  The time difference, in hours, between the PLC time and the UTC time.
  - The value range is from +12 hours to -12 hours.
  - The default value is 0.

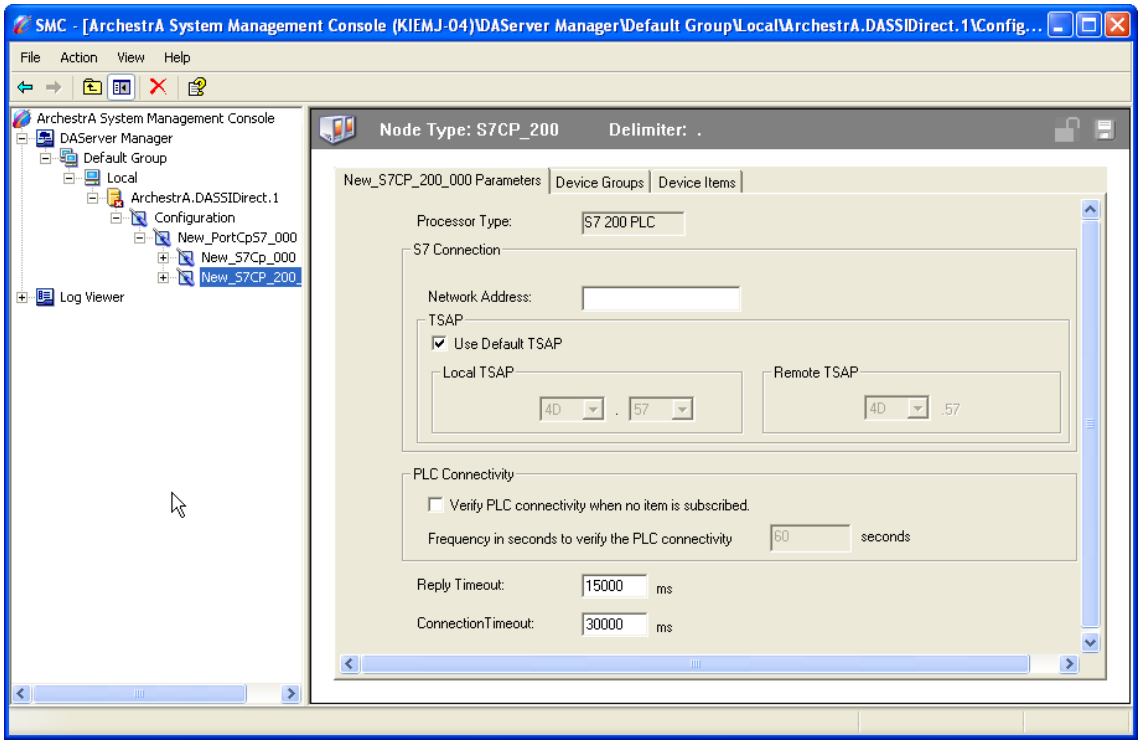
S7Cp_200 Object

The SIDirect DAServer can connect to different CPs in an S7-200 PLC. Each of the S7Cp_200 nodes models the end-point of the communications path.

From the New_PortCpS7_000 branch of the DAServer hierarchy, create the S7Cp_200 object.

To add S7Cp-200 object to your SIDirect hierarchy

1 Select and right-click your **New_PortCpS7_000** object, then select **Add S7Cp_200 Object**. The New_S7Cp_200_000 Parameters configuration view appears.



- A maximum of 1024 S7Cp_200 objects can be created from the **New_PortCpS7** branch.

2 Rename as appropriate.

The S7Cp_200 object configuration view has five elements, four of which are configurable:

- **Processor Type**
  The processor type is S7-200 PLC.
- **S7 Connection**
  The S7 Connection has the following three configurable parameters.
  - **Network Address**
    The IP address or host name of the remote S7-200 CP. Type in the network address where the PLC is located (for example, "10.11.12.13"), or a host name if one is defined in the local hostlist. The length of the address cannot be more than 255 characters. The first and last character must be alphanumeric. The rest of the characters must be either alphanumeric, "-" (dash), or "." (period).

- **Local TSAP**
  The local TSAP of the computer. The Local TSAP consists of two (2) Hex numbers. The first number corresponds to the connection resource. Each number ranges from 00 to FF. The default values are 4D and 57, respectively.
- **Remote TSAP**
  This Remote TSAP corresponds to what you configured in the TSAP for the S7-200 CP. The Remote TSAP consists of two (2) Hex numbers. Each number ranges from 00 to FF. The default values are 4D and 57, respectively.

- **PLC Connectivity**
  The watchdog scheme for detecting the connectivity status to the PLC when there are no activities (no items subscribed to).
  - Select the **Verify PLC connectivity when no item is subscribed** check box to turn on the watchdog.
  - Specify the watchdog time interval, in seconds, in the **Frequency in seconds to verify the PLC connectivity** box.
- **Reply Timeout**
  Enter a value, in milliseconds, beyond which messages time out.
  - Allowable range is 0 to 100,000 milliseconds.
  - The default value is 15,000 milliseconds.

  If you decrease this value, the SIDirect DAServer reacts faster to a communications failure.
- **Connection Timeout**
  Enter a value, in milliseconds, beyond which a pending request to initiate a connection times out.
  - Allowable range is 0 to 100,000 milliseconds.
  - The default value is 30,000 milliseconds.

The logical endpoint for each branch of the SIDirect hierarchy tree is a hardware device (PLC).

Note When adding a hierarchy object, the default name is in the format of **New_ObjectName_###**. ObjectName is the name of the object type and ### is a numeric value starting from "000" 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 the PortCpS7 root of this DAServer down to the leaf.
This creates a link name that is always unique for the DAServer.

To use the DAServer, you have to activate it. See the DAServer Manager documentation for information about activating and deactivating the DAServer.

Configuring Device Group and Device Item Definitions

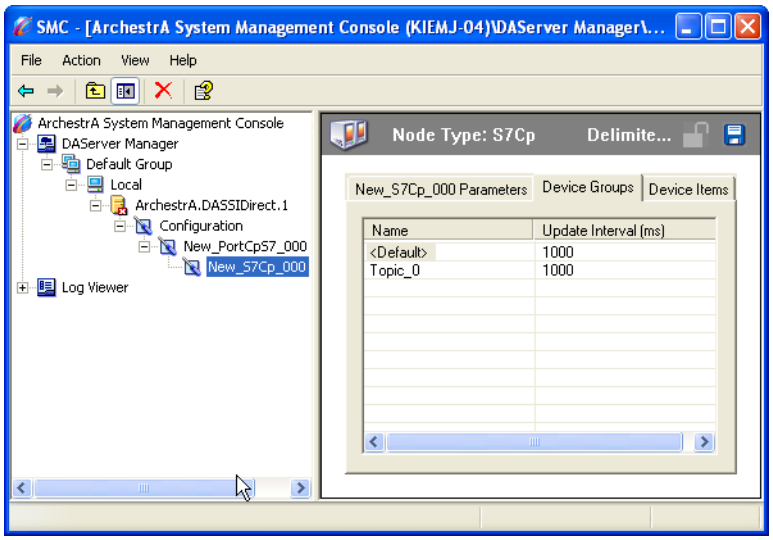
Select the **Device Groups** tab or the **Device Items** tab in the DAServer Manager user interface to create new, modify, or delete device-group and device-item definitions for an object.

- For DDE/SuiteLink communications, one or more device-group definitions must exist for each PLC that the SIDirect DAServer communicates with.
- Each device-group (topic) definition must contain a unique name for the PLC associated with it.

Each configuration view associated with objects in the SIDirect DAServer hierarchy tree has a **Save** button. When you modify the **Parameters**, **Device Groups** dialog box, or the **Device Items** dialog box, click **Save** to implement the new modifications. If you try to open another configuration dialog box you are prompted to save the new data to the configuration set.

Device Group Definitions

The **Device Groups** dialog box, which appears by clicking the **Device Groups** tab in the **New_S7Cp_000 Parameters** or the **New_S7Cp_200_000 Parameters** configuration view, is the place where you create, add, delete, and define device groups. You can also configure default update intervals for the objects and edit update intervals in this dialog box.



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

The device group for the S7CP_200 object is not editable.

To create or add device groups

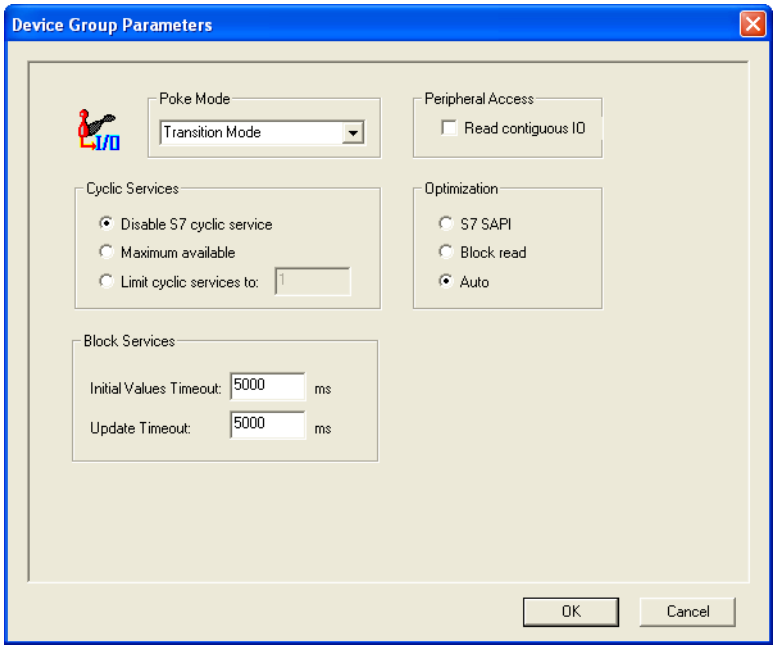
1. Right-click in the **Device Groups** box and click **Add**.
2. Enter a unique name for the device group. Device group names are case-insensitive.

To delete device groups

- Right-click on the device group to be deleted from the list and select **Delete**.

To make edits on device groups

1 Make edits on a device group by right-clicking on the device group to be edited. Select **Edit** from the menu to open the **Device Group Parameters** dialog box.



2 Make the necessary edits.

The **Edit** dialog box contains five configurable elements:

- **Poke Mode**
  Select one of three settings to tune the poking behavior to the PLC.
  - **Control mode**
    This mode preserves the poke order and does not fold write values. Select this mode when using this device group with control clients such as the InBatch and InControl applications. If selected, the server processes all poked values in the order they are received from a client and does not discard any poke values even when several values are poked to the same item.
  - **Transition mode**
    This mode preserves the poke order but allows folding of poke values in the following way: if the server receives more than one value per item it can discard poke values except for the first, second, and last value for this item.
    Transition mode prevents InTouch sliders from stuttering.

- **Full optimization**
  This mode allows changing the poke order and folding of poke values by poking the last value of a series of pokes to one item only. This minimizes bus traffic and poke duration. Use this mode for high volume pokes such as recipe downloads where the sequence of poke does not matter.

Note The highest performance can be achieved with the Full optimization mode. Consider your data consistency requirements with respect to poke order and folding.

The default Poke Mode settings for the SIDirect DAServer is Transition mode.

- **Peripheral Access**
  Select the **Read contiguous IO** check box to force the SIDirect DAServer to read input and output blocks (also peripherals) if their address spaces are not contiguous. Select this check box if you have some holes in the input-address or output-address space in your PLC.

- **Cyclic Services**
  If you use Cyclic Services (the **Disable S7 cyclic service** option is **not** selected), configure two additional settings:

  - Maximum available
  - Limit cyclic services to

  If you know how many services the remote PLC can handle, you can limit the use of cyclic services in this device group and distribute the available cyclic services among the device groups associated with this connection. Otherwise, you can use the maximum available services.

  - Allowable range for **Limit cyclic services to** is 1 to 150.
  - The default is 1.

  Click on **Disable S7 cyclic service** to disable the S7 cyclic services for the device group.

  - Cyclic services have a reliable update frequency and need less bus access. They are a limited resource in the PLC and/or Communications Processor.

  If you select the **Disable S7 cyclic service** option, the SIDirect DAServer polls all topics in this device group. If you do not select this option, the SIDirect DAServer tries to register as many topics in the cyclic services as possible.

- If there are insufficient credits for cyclic services left, the SIDirect DAServer creates poll messages instead of cyclic messages for the remaining items.

Reasons for disabling Cyclic Services include the following:

- There is a device group with a long update interval. It should not occupy the cyclic services resource.
- You need to force the SIDirect DAServer to attempt to collect data faster than 100 milliseconds.
- You activate and deactivate items very often.

- **Optimization**
  Select one of the following settings to configure the optimization mode the SIDirect DAServer needs to use to acquire data from the PLC:
  - S7 SAPI
  - Block read
  - Auto

  The default is Auto. For more detailed information, see Optimization Considerations on page 35.
- **Block Services**
  If the Block Services function is required, you have to configure two settings in the **Block Services** box:
  - **Initial Values Timeout**
    Allowable range for Initial Values Timeout is 0 to 65,535 milliseconds; the default value is 5,000.
  - **Update Timeout**
    Allowable range for Update Timeout is 0 to 65,535 milliseconds. The default value is 5,000.

  The Block Services function needs time-outs to supervise reading initial values and updating the block items to this connection. A time-out value of 0 (zero) disables the time supervision of block messages.

  Block services are unconfirmed services. If the remote station does not send data within this time range, the Block Services is reinitialized and an error message appears in the WWLogger.

To configure default update intervals

- To configure a default update interval for the object, right-click in the **Device Groups** box and click **Config Default Update Interval** on the 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 edits.
  - Update Interval is the frequency in milliseconds that the SIDirect 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 SIDirect DAServer tree hierarchy, you are prompted to save the modifications to the configuration set.

Optimization Considerations

The SIDirect DAServer uses the following same optimization considerations as the S7 I/O Server:

- Use different Poke modes.
- Use different reading optimization modes.
- Use cyclic services to minimize traffic.
- Use block services to minimize traffic.

The DAServer can also optimize its performance in getting data from the PLC by using the optimization mode. The following options are available:

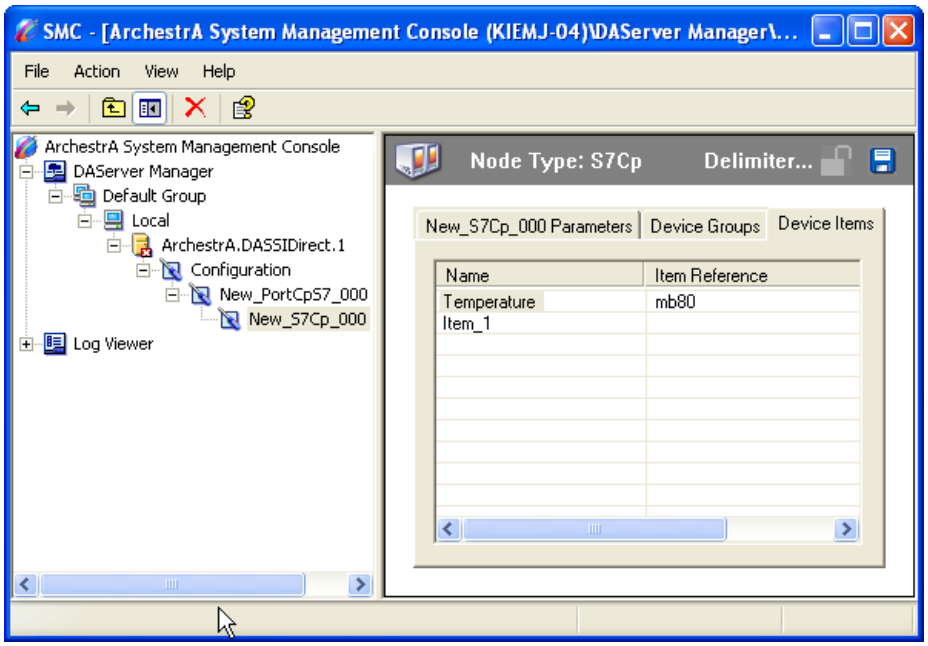
- S7 SAPI
  The S7 SAPI mode is the same optimization mode used in the pre-release of the former Wonderware Siemens SIMATIC NET S7 I/O Server. This mode is implemented to keep the server compatible to the former server. This mode is the less-preferred optimization mode.
- Block Read
  The Block Read mode always registers a whole byte array containing some items. If you frequently switch items (activating and deactivating) that have similar addresses, this mode is the best selection. In this mode, there are less activations and deactivations on the protocol.
- Auto
  By default, the Auto mode is best to use when exploiting the whole PDU. The Auto mode has the best performance in cases where you do not make many activations and deactivations.

Device Item Definitions

The predefined item syntax/name for the S7 PLC cannot be changed. However, to make it easier to remember item names, you can create aliases for these item names.
For example, it may be easier for you to remember the item syntax "mb80" as "Temperature."

Note The SIDirect DAServer only supports 1,000 aliases.

Select the **Device Items** tab in the DAServer Manager user interface to create new, modify, delete, export, or import device-item definitions for an object. The configuration is performed in the **Device Items** dialog box, which appears when you click the **Device Items** tab in the **New_S7Cp_000 Parameters** configuration view.



After you configure item names, the DAServer can perform OPC Item browsing. When the DAServer is running and an OPC client requests item information, the configured items appear under the PLC hierarchy node. User-defined data types appears at the lowest level in the hierarchy when browsed via the OPC client.

To create or add device items

1. Right-click in the **Device Items** box and click **Add**.
2. Type in the item name of your choice in the **Name** column. For example, "Temperature."
3. When you add a new device item, enter a unique name.

4 Double-click the line in the **Item Reference** column and enter the correlated item reference for the name you selected.
For example, "mb80."

Note If the name and the item reference are the same, it is only necessary to enter a name. The DAServer assumes that the item reference is the same. This is necessary if you want to add some items for browsing via the OPC, even if they do not have a symbolic name.

To rename device items

- Right-click on the device item to be renamed and click **Rename**. Make the changes.

To delete device items

- Right-click on the device item to be deleted from the list and click **Delete**.

To clear all device items

- Right-click in the **Device Items** box and click **Clear All**. All the device items listed are cleared after you confirm their deletion.

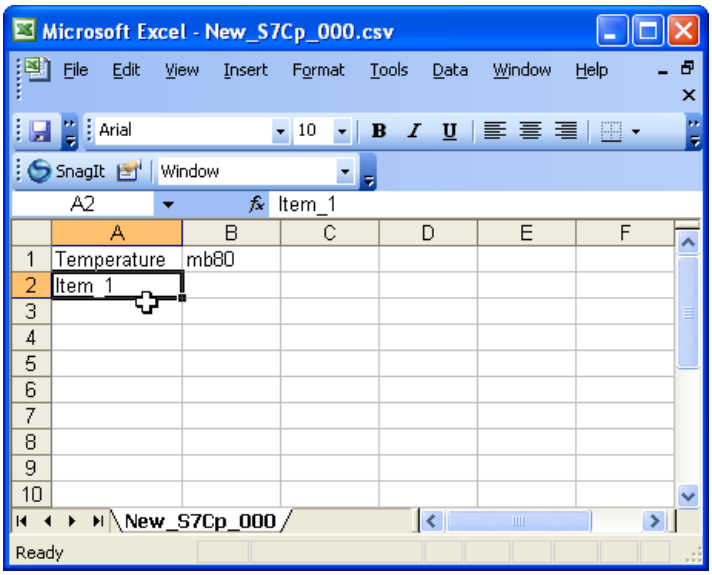
Exporting and Importing DAServer Item Data

You can export and import the DAServer item data to and from a CSV file after you configure the Device Items. This lets you perform an off-line, large-scale edit on the item data configured for a PLC and import that data back into the PLC configuration.

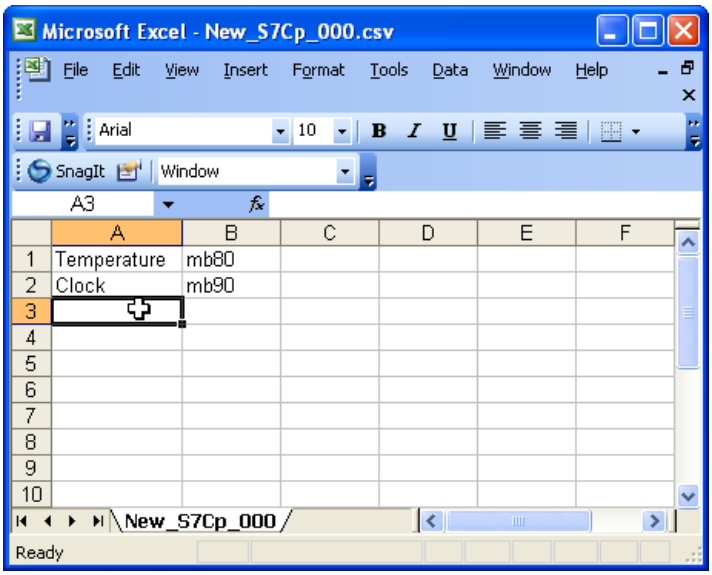
To export DAServer item data to a CSV file

1 Right-click in the **Device Items** box and click **Export**. The **Save As** dialog box appears and the file name defaults to "PLC Hierarchyname.csv," within the current-system-configured default directory.

2 Accept the defaults to save the file. The file is saved as New_S7Cp_000.csv. Now you can edit it in Microsoft Excel.



The file contains one row for each item configured with two columns: Name and Item Reference.



After you are done editing, you are ready to import the edited file into the DAServer Manager.

To import DAServer item data from a CSV file

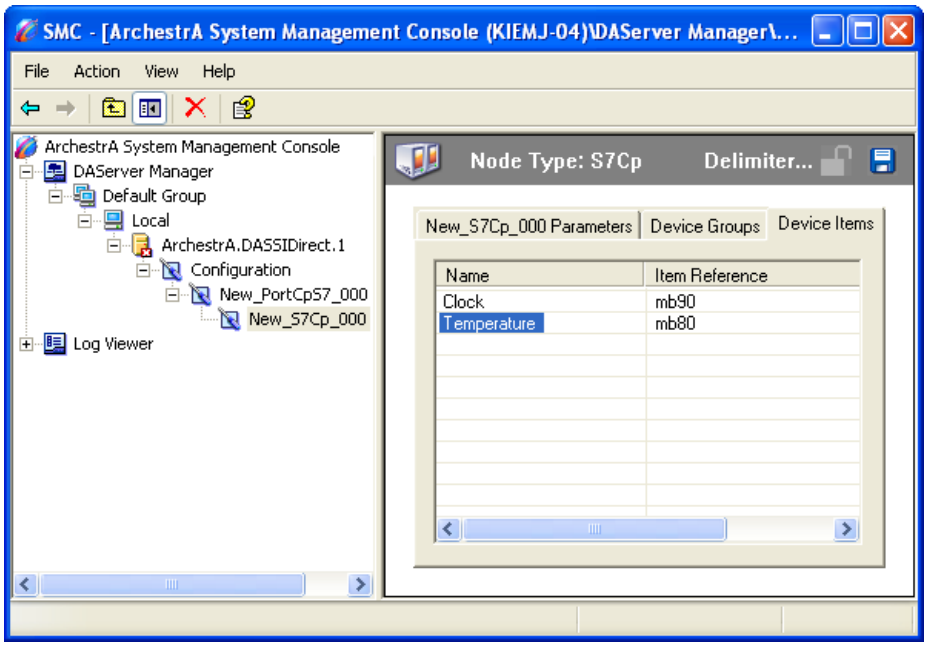
1 Right-click in the **Device Items** box.

2 Click **Clear All** to clear all the item data you want to replace with the edited .csv file. The data is cleared after you click on **Yes** to confirm the deletion.

3 Click **Import** on the menu. The **Open** dialog box appears.

4 Browse for the .csv file you want to import, select it, then click **OK**.
The DAServer Manager imports the file. You see the data in the **Device Items** box.

When the file is imported, new item references are added based on unique names. If there are duplicate names, you can replace the existing entry with a new entry or ignore the new entry.



When the DAServer is running and an OPC Client requests item information, the imported configured items appear under the PLC hierarchy node.

Note The SIDirect DAServer does not support importing preconfigured items (alias names of items that work with OPC) in an output file generated by the Siemens Step7 software.

Scan-Based Message Handling

Wonderware DAServers poll hardware devices for information. This polling is requested by one or more clients.

After a client requests a particular piece of information, the SIDirect DAServer creates its own request and sends that request to the hardware device. The SIDirect DAServer then waits for a response to its request. After the SIDirect DAServer receives the information, it passes that information back to the client and repeats the process until all clients stop requesting information.

You define the rate at which the SIDirect DAServer polls a particular device for a specific piece of information in the device group (topic definition) inside the SIDirect DAServer. You use 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.

Because 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 item is polled at this rate, the entire system suffers due to slow response time. Therefore, 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.

Unsolicited Message Handling

The SIDirect DAServer processes the following three types of unsolicited messages sent by the S7 PLCs:

- Alarms
- Events
- Block Services

Alarms and Events

In the world of PLCs and DAServers, it is obvious that a PLC knows when a critical event occurs before the DAServer polls for that data. Therefore, it seems natural that if a critical event occurs, the PLC can inform the DAServer immediately, without waiting for the DAServer to poll it.

This is the role of an unsolicited message. After a PLC determines that a critical condition exists, it can generate a message immediately sent to the DAServer without a previous request from the DAServer. The unsolicited message implementation requires both the messaging instructions properly programmed in the PLC logic and the device group appropriately configured in the DAServer.

Block Services

In addition to unsolicited messages based on critical conditions or events, S7 PLCs can also handle another type of unsolicited messages called Block Services.

You can use Block Services to send blocks of data up to 64KBytes within one send job and trigger it by using a timer, an event, an I/O activity, or initiate it via a program code.

Archiving Configuration Sets

After you configure your DAServer, you can archive that specific configuration. You can archive more than one configuration set, and select 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. Click **Archive Configuration Set**.
2. In the **Archive Configuration Set** dialog box, provide a Configuration Set Name.
3. Click **Archive**. All current configuration values are saved to the archived set.

After you archive 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 selected configuration set.

Note If you do not explicitly archive the current configuration, it is automatically saved in a default configuration called "dassidirect."

Hot Configuration

If a parameter value change takes effect right away while the DAServer is running, the parameter is a hot-configurable parameter. Certain parameters in the SIDirect DAServer are 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, the **Update Interval** column in the **Device Groups** tab, and device items.

All other parameters are not hot-configurable. To make those changes take effect, restart the DAServer.

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

PLC Connectivity Detection

The SIDirect DAServer provides a watchdog scheme for detecting the connectivity status to the PLC when there are no activities (no items subscribed to).

- You can turn the watchdog on and off when configuring the S7Cp and S7Cp_200 objects.
- You can specify the watchdog time interval, in seconds, to verify the PLC conectivity.
- The SIDirect DAServer internally sends a message to read the PLC based on the watchdog interval when there are no activities (no items subscribed to) in the noncyclic topics.
- The SIDirect DAServer then updates the value of the system item SYSStatus according to the condition of the PLC connectivity.

Chapter 3

Item Naming

The Wonderware SIDirect DAServer uses an item-naming convention based on the two-letter data-type identifiers used in programming the Siemens PLCs. The server accepts both the English and German standard identifiers.

The tables in this section describe the item naming convention for the Siemens S7 PLCs. The ranges specified in those tables vary according to the type of the controller used.

Data Blocks and Instance Blocks

The following table summarizes the data format, item or point, suffix, data type, and range for Data Blocks and Instance Blocks.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	D<B,I>*d*,X*x*.*y*		VT_BOOL	0 or 1
String	D<B,I>*d*,S*x*,*v*		VT_BSTR	String
	D<B,I>*d*,STRING*x*,*v*		VT_BSTR	String
S7String	D<B,I>*d*,S7S*x*,*w*		VT_BSTR	String
	D<B,I>*d*,S7STRING*x*,*w*		VT_BSTR	String
Byte	D<B,I>*d*,B*x*		VT_UI1	0 to 255
	D<B,I>*d*,BYTE*x*		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Byte Array	D<B,I>*d*,B*x*,*v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	D<B,I>*d*,BYTE*x*,*v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
Char	D<B,I>*d*,CHAR*x*		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999

Data Format	Item/Point	Suffix	Data Type	Range
Char Array	D<B,I>*d*,CHAR*x*,*v*		VT_ARRAY|VT_UI1	-128 to 127 for each element*
Word	D<B,I>*d*,W*n*		VT_UI2	0 to 65535
	D<B,I>*d*,WORD*n*		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31
Word Array	D<B,I>*d*,W*n*,*v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	D<B,I>*d*,WORD*n*,*v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
Integer	D<B,I>*d*,INT*n*		VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	D<B,I>*d*,INT*n*,*v*		VT_ARRAY|VT_UI1	-32768 to 32767 for each element*
Double Word	D<B,I>*d*,D*m*		VT_UI4	0 to 4294967295**
	D<B,I>*d*,DWORD*m*		VT_UI4	0 to 4294967295**
		BCD	VT_UI4	0 to 99999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648 MS to 24D_20H_31M_23S_647MS
Double Word Array	D<B,I>*d*,D*m*,*v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	D<B,I>*d*,DWORD*m*,*v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
Double Integer	D<B,I>*d*,DINT*m*		VT_I4	-2147483648 to 2147483647
		BCD	VT_I4	-9999999 to 9999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648 MS to 24D_20H_31M_23S_647MS
Double Integer Array	D<B,I>*d*,DINT*m*,*v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element*
Real	D<B,I>*d*,REAL*m*		VT_R4	±3.4e38
Real Array	D<B,I>*d*,REAL*m*,*v*		VT_ARRAY|VT_R4	±3.4e38 for each element*

*: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big endian format when data is returned to the DDE/SuiteLink layer.

**: For DDE/SuiteLink, this value is restricted to the range of 0 to 2147483647. Values higher than that are clamped to the maximum value of 2147483647 in a SuiteLink or DDE client. In this case, the quality of the item shows "Clamp High."

Where:

d is the data block number, with a range from 1 to 65535.

x is the start address. For S7-300/400 PLCs, the range is from 0 to 65535. For S7-200 PLCs, the value is 1.

y is the bit position, with a range from 0 to 7.

- 0 is the LSB (Least Significant Bit).
- 7 is the MSB (Most Significant Bit).

n is the start address of 2-byte data/2-byte data arrays.

m is the start address of 4-byte data/4-byte data arrays.

v is the length of data in bytes, with a range from 0 to 65534.

w is the length of the net S7 string data in characters (size in S7 message is w+1, size of string representation in S7 PLC is w+2).

Note All data blocks are **Read/Write**. The longest string or array that can be read in a cyclic service has the length of the PDU size minus 32 bytes. The longest string the InTouch software can process is 131 bytes. The longest string that can be poked is 256 bytes or the PDU size minus 28 bytes, whichever is less. The SIDirect DAServer processes a write (**POKE**) to a Data Block.

Examples for S7-300/400 PLCs:

DB123,W24

DB23,DINT10BCD

DI5,X2.0

DI6,BYTE4,10

Flag Bytes

The following table summarizes data format, item or point, suffix, data type, and range for Flag Bytes.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	FX*x.y*		VT_BOOL	0 or 1
	MX*x.y*		VT_BOOL	0 or 1
String	FS*x,v*		VT_BSTR	String
	MS*x,v*		VT_BSTR	String
	FSTRINGx,v		VT_BSTR	String
	MSTRING*x,v*		VT_BSTR	String
Byte	FB*x*		VT_UI1	0 to 255
	MB*x*		VT_UI1	0 to 255
	FBYTE*x*		VT_UI1	0 to 255
	MBYTE*x*		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Byte Array	FB*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	MB*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	FBYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	MBYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
Char	FCHAR*x*		VT_I1	-128 to 127
	MCHAR*x*		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Char Array	FCHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
	MCHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
Word	FW*n*		VT_UI2	0 to 65535
	MW*n*		VT_UI2	0 to 65535
	FWORDn		VT_UI2	0 to 65535
	MWORD*n*		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31

Data Format	Item/Point	Suffix	Data Type	Range
Word Array	FW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	MW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	FWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	MWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
Integer	FINT*n*		VT_I2	-32768 to 32767
	MINT*n*		VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	FINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
	MINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
Double Word	FD*m*		VT_UI4	0 to 4294967295**
	MD*m*		VT_UI4	0 to 4294967295**
	FDWORD*m*		VT_UI4	0 to 4294967295**
			VT_UI4	0 to 4294967295**
	MDWORD*m*	BCD	VT_UI4	0 to 9999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Word Array	FD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	MD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	FDWORDm,v		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	MDWORD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
Double Integer	FDINT*m*		VT_14	-2147483648 to 2147483647
	MDINT*m*		VT_14	-2147483648 to 2147483647
		BCD	VT_14	-9999999 to 9999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648 MS to 24D_20H_31M_23S_647MS

Data Format	Item/Point	Suffix	Data Type	Range
Double Integer Array	FDINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element*
	MDINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element*
Real	FREAL*m*		VT_R4	±3.4e38
	MREAL*m*		VT_R4	±3.4e38
Real Array	FREAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element*
	MREAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element*

*: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

**: For DDE/SuiteLink, this value is restricted to the range of 0 to 2147483647. Values higher than that are clamped to the maximum value of 2147483647 in a SuiteLink or DDE client. In this case, the quality of the item shows "Clamp High."

Where:

x	is the start address, with a range from 0 to 65535.
y	is the bit position, with a range from 0 to 7. • 0 is the LSB (Least Significant Bit). • 7 is the MSB (Most Significant Bit).
n	is the start address of 2-byte data/2-byte data arrays, with a range from 0 to 65534.
m	is the start address of 4-byte data/4-byte data arrays, with a range from 0 to 65532.
v	is the length of data in bytes, with a range from 1 to (net PDU data size/type size - header information).

Note All flags are **Read/Write** (*x*=0 to 65535, *y*=0 to 7, *n*=0 to 65534, *m*=0 to 65532, *v*=1 to net PDU data size/type size - header information, this size may vary). The longest string or array that can be read in a cyclic service is the length of the PDU size minus 32 bytes. The longest string the InTouch software can process is 131 bytes. The longest string that can be poked is 256 bytes or the PDU size minus 28 bytes, whichever is less. The SIDirect DAServer processes a write (**POKE**) to a Flag Byte.

Input Bytes

The following table summarizes the data format, item or point, suffix, data type, and range for Input Bytes.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	I*x.y*		VT_BOOL	0 or 1
	E*x.y*		VT_BOOL	0 or 1
	IX*x.y*		VT_BOOL	0 or 1
	EX*x.y*		VT_BOOL	0 or 1
String	IS*x,v*		VT_BSTR	String
	ES*x,v*		VT_BSTR	String
	ISTRING*x,v*		VT_BSTR	String
	ESTRING*x,v*		VT_BSTR	String
Byte	IB*x*		VT_UI1	0 to 255
	EB*x*		VT_UI1	0 to 255
	IBYTE*x*		VT_UI1	0 to 255
	EBYTE*x*		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Byte Array	IB*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	EB*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	IBYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	EBYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
Char	ICHAR*x*		VT_I1	-128 to 127
	ECHAR*x*		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Char Array	ICHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
	ECHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*

Data Format	Item/Point	Suffix	Data Type	Range
Word	IW*n*		VT_UI2	0 to 65535
	EW*n*		VT_UI2	0 to 65535
	IWORD*n*		VT_UI2	0 to 65535
	EWORD*n*		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31
Word Array	IW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	EW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	IWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	EWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
Integer	IINT*n*		VT_I2	-32768 to 32767
	EINT*n*		VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	IINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
	EINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
Double Word	ID*m*		VT_UI4	0 to 4294967295**
	ED*m*		VT_UI4	0 to 4294967295**
	IDWORD*m*		VT_UI4	0 to 4294967295**
	EDWORD*m*		VT_UI4	0 to 4294967295**
		BCD	VT_UI4	0 to 99999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648 MS to 24D_20H_31M_23S_647 MS

Data Format	Item/Point	Suffix	Data Type	Range
Double Word Array	ID*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	ED*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element
	IDWORD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	EDWORD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element
Double Integer	IDINT*m*		VT_I4	-2147483648 to 2147483647
	EDINT*m*		VT_I4	-2147483648 to 2147483647
		BCD	VT_I4	-9999999 to 9999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_ 648 MS to 24D_20H_31M_23S_647MS
Double Integer Array	IDINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element*
	EDINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element*
Real	IREAL*m*		VT_R4	±3.4e38
	EREAL*m*		VT_R4	±3.4e38
Real Array	IREAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element*
	EREAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element

*: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

**: For DDE/SuiteLink, this value is restricted to the range of 0 to 2147483647. Values higher than that are clamped to the maximum value of 2147483647 in a SuiteLink or DDE client. In this case, the quality of the item shows "Clamp High."

Where:

x is the start address, with a range from 0 to 65535.

y is the bit position, with a range from 0 to 7.

- 0 is the LSB (Least Significant Bit).
- 7 is the MSB (Most Significant Bit).

Where:

n	is the start address of 2-byte data/2-byte data arrays, with a range from 0 to 65534.
m	is the start address of 4-byte data/4-byte data arrays, with a range from 0 to 65532.
v	is the length of data in bytes, with a range from 1 to (net PDU data size/type size - header information).

Note All inputs are **Read-Only**. The longest string or array that can be read in a cyclic service is the length of the PDU size minus 32 bytes. The longest string the InTouch software can process is 131 bytes. The SIDirect DAServer does **not** process a write (**POKE**) to an Input Byte.

Output Bytes

The following table summarizes data format, item or point, suffix, data type, and range for Output Bytes.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	O*x.y*		VT_BOOL	0 or 1
	A*x.y*		VT_BOOL	0 or 1
	Q*x.y*		VT_BOOL	0 or 1
	OX*x.y*		VT_BOOL	0 or 1
	AX*x.y*		VT_BOOL	0 or 1
	QX*x.y*		VT_BOOL	0 or 1
String	OS*x,v*		VT_BSTR	String
	AS*x,v*		VT_BSTR	String
	QS*x,v*		VT_BSTR	String
	OSTRING*x,v*		VT_BSTR	String
	ASTRING*x,v*		VT_BSTR	String
	QSTRING*x,v*		VT_BSTR	String

Data Format	Item/Point	Suffix	Data Type	Range
Byte	OB*x*		VT_UI1	0 to 255
	AB*x*		VT_UI1	0 to 255
	QB*x*		VT_UI1	0 to 255
	OBYTE*x*		VT_UI1	0 to 255
	ABYTE*x*		VT_UI1	0 to 255
	QBYTE*x*		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Byte Array	OB*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	AB*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	QB*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	OBYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	ABYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	QBYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
Char	OCHAR*x*		VT_I1	-128 to 127
	ACHAR*x*		VT_I1	-128 to 127
	QCHAR*x*		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Char Array	OCHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
	ACHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
	QCHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
Word	OW*n*		VT_UI2	0 to 65535
	AW*n*		VT_UI2	0 to 65535
	QW*n*		VT_UI2	0 to 65535
	OWORD*n*		VT_UI2	0 to 65535
	AWORD*n*		VT_UI2	0 to 65535
	QWORD*n*		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31

Data Format	Item/Point	Suffix	Data Type	Range
Word Array	OW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	AW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element
	QW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element
	OWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	AWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element
	QWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element
Integer	OINT*n*		VT_I2	-32768 to 32767
	AINT*n*		VT_I2	-32768 to 32768
	QINT*n*		VT_I2	-32768 to 32768
		BCD		-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	OINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
	AINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
	QINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
Double Word	OD*m*		VT_UI4	0 to 4294967295**
	AD*m*		VT_UI4	0 to 4294967295**
	QD*m*		VT_UI4	0 to 4294967295**
	ODWORD*m*		VT_UI4	0 to 4294967295**
	ADWORD*m*		VT_UI4	0 to 4294967295**
	QDWORD*m*		VT_UI4	0 to 4294967295**
		BCD	VT_UI4	0 to 99999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS

Data Format	Item/Point	Suffix	Data Type	Range
Double Word Array	OD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	AD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	QD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	ODWORD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	ADWORD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	QDWORD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
Double Integer	ODINT*m*		VT_I4	-2147483648 to 2147483647
	ADINT*m*			
	QDINT*m*			
		BCD	VT_I4	-9999999 to 9999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Integer Array	ODINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element*
	ADINT*m,v*			
	QDINT*m,v*			
Real	OREAL*m*		VT_R4	±3.4e38
	AREAL*m*			
	QREAL*m*			
Real Array	OREAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element*
	AREAL*m,v*			
	QREAL*m,v*			

*: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

**: For DDE/SuiteLink, this value is restricted to the range of 0 to 2147483647. Values higher than that are clamped to the maximum value of 2147483647 in a SuiteLink or DDE client. In this case, the quality of the item shows "Clamp High."

Where:

x is the start address, with a range from 0 to 65535.

y is the bit position, with a range from 0 to 7.

- 0 is the LSB (Least Significant Bit).
- 7 is the MSB (Most Significant Bit).

n is the start address of 2-byte data/2-byte data arrays, with a range from 0 to 65534.

m is the start address of 4-byte data/4-byte data arrays, with a range from 0 to 65532.

v is the length of data in bytes, with a range from 1 to (net PDU data size/type size - header information).

Note All outputs are **Read/Write**. The longest string or array that can be read in a cyclic service is the length of the PDU size minus 32 bytes. The longest string the InTouch software can process is 131 bytes. The longest string that can be poked is 256 bytes or the PDU size minus 28 bytes, whichever is less. The SIDirect DAServer processes a write (**POKE**) to an Output Byte.

Peripheral Input Bytes

The following table summarizes the data format, item or point, suffix, data type, and range for Peripheral Input Bytes.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	PI*x.y*		VT_BOOL	0 or 1
	PE*x.y*		VT_BOOL	0 or 1
	PIX*x.y*		VT_BOOL	0 or 1
	PEX*x.y*		VT_BOOL	0 or 1
String	PIS*x,v*		VT_BSTR	String
	PES*x,*v		VT_BSTR	String
	PISTRING*x,v*		VT_BSTR	String
	PESTRING*x,v*		VT_BSTR	String

Data Format	Item/Point	Suffix	Data Type	Range
Byte	PIB*x*		VT_UI1	0 to 255
	PEB*x*		VT_UI1	0 to 255
	PIBYTE*x*		VT_UI1	0 to 255
	PEBYTE*x*		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Byte Array	PIB*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	PEB*x,*v		VT_ARRAY|VT_UI1	0 to 255 for each element*
	PIBYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	PEBYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
Char	PICHAR*x*		VT_I1	-128 to 127
	PECHAR*x*		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Char Array	PICHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
	PECHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
Word	PIW*n*		VT_UI2	0 to 65535
	PEW*n*		VT_UI2	0 to 65535
	PIWORD*n*		VT_UI2	0 to 65535
	PEWORD*n*		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31
Word Array	PIW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	PEW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	PIWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	PEWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
Integer	PIINT*n*		VT_I2	-32768 to 32767
	PEINT*n*		VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	PIINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
	PEINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*

Data Format	Item/Point	Suffix	Data Type	Range
Double Word	PID*m*		VT_UI4	0 to 4294967295**
	PED*m*		VT_UI4	0 to 4294967295**
	PIDWORD*m*		VT_UI4	0 to 4294967295**
	PEDWORD*m*		VT_UI4	0 to 4294967295**
		BCD	VT_UI4	0 to 99999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Word Array	PID*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	PED*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element
	PIDWORD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element
	PEDWORD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element
Double Integer	PIDINT*m*		VT_I4	-2147483648 to 2147483647
	PEDINT*m*		VT_I4	-2147483648 to 2147483647
		BCD	VT_I4	-9999999 to 9999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Integer Array	PIDINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element*
	PEDINT*m,v*			
Real	PIREAL*m*		VT_R4	±3.4e38
	PEREAL*m*		VT_R4	±3.4e38
Real Array	PIREAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element*
	PEREAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element

*: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

**: For DDE/SuiteLink, this value is restricted to the range of 0 to 2147483647. Values higher than that are clamped to the maximum value of 2147483647 in a SuiteLink or DDE client. In this case, the quality of the item shows "Clamp High."

The S7-200 PLCs do **not** support Peripheral Input Bytes.

Where:

x	is the start address, with a range from 0 to 65535.
y	is the bit position, with a range from 0 to 7. • 0 is the LSB (Least Significant Bit). • 7 is the MSB (Most Significant Bit).
n	is the start address of 2-byte data/2-byte data arrays, with a range from 0 to 65534.
m	is the start address of 4-byte data/4-byte data arrays, with a range from 0 to 65532.
v	is the length of data in bytes, with a range from 1 to (net PDU data size/type size - header information).

Note All peripheral inputs are **Read-Only**. The longest string or array that can be read in a cyclic service is the length of the PDU size minus 32 bytes. The longest string the InTouch software can process is 131 bytes. The longest string that can be poked is 256 bytes or the PDU size minus 28 bytes, whichever is less. The SIDirect DAServer does **not** process a write (**POKE**) to a Peripheral Input Byte. Some input modules are not readable.

Peripheral Output Bytes

The following table summarizes the data format, item or point, suffix, data type, and range for Peripheral Output Bytes.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	PO*x.y*		VT_BOOL	0 or 1
	PA*x.y*		VT_BOOL	0 or 1
	PQ*x.y*		VT_BOOL	0 or 1
	POX*x.y*		VT_BOOL	0 or 1
	PAX*x.y*		VT_BOOL	0 or 1
	PQX*x.y*		VT_BOOL	0 or 1
String	POS*x,v*		VT_BSTR	String
	PAS*x,v*		VT_BSTR	String
	PQS*x,v*		VT_BSTR	String
	POSTRING*x,v*		VT_BSTR	String
	PASTRING*x,v*		VT_BSTR	String
	PQSTRING*x,v*		VT_BSTR	String
Byte	POB*x*		VT_UI1	0 to 255
	PAB*x*		VT_UI1	0 to 255
	PQB*x*		VT_UI1	0 to 255
	POBYTE*x*		VT_UI1	0 to 255
	PABYTE*x*		VT_UI1	0 to 255
	PQBYTE*x*		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Byte Array	POB*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	PAB*x,*v		VT_ARRAY|VT_UI1	0 to 255 for each element*
	PQB*x,*v		VT_ARRAY|VT_UI1	0 to 255 for each element*
	POBYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	PABYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	PQBYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
Char	PACHAR*x*		VT_I1	-128 to 127
	POCHAR*x*		VT_I1	-128 to 127
	PQCHAR*x*		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999

Data Format	Item/Point	Suffix	Data Type	Range
Char Array	POCHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
	PACHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
	PQCHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
Word	POW*n*		VT_UI2	0 to 65535
	PAW*n*		VT_UI2	0 to 65535
	PQW*n*		VT_UI2	0 to 65535
	POWORD*n*		VT_UI2	0 to 65535
	PAWORD*n*		VT_UI2	0 to 65535
	PQWORD*n*		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31
Word Array	POW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	PAW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	PQW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	POWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	PAWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	PQWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
Integer	POINT*n*		VT_I2	-32768 to 32767
	PAINT*n*		VT_I2	-32768 to 32767
	PQINT*n*		VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	POINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
	PAINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
	PQINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*

Data Format	Item/Point	Suffix	Data Type	Range
Double Word	POD*m*		VT_UI4	0 to 4294967295**
	PAD*m*		VT_UI4	0 to 4294967295**
	PQD*m*		VT_UI4	0 to 4294967295**
	PODWORD*m*		VT_UI4	0 to 4294967295**
	PADWORD*m*		VT_UI4	0 to 4294967295**
	PQDWORD*m*		VT_UI4	0 to 4294967295**
		BCD	VT_UI4	0 to 99999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Word Array	POD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	PAD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	PQD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
Double Integer	PODINT*m*		VT_I4	-2147483648 to 2147483647
	PADINT*m*		VT_I4	-2147483648 to 2147483647
	PQDINT*m*		VT_I4	-2147483648 to 2147483647
		BCD	VT_I4	-9999999 to 9999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Integer Array	PODINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element*
	PADINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element*
	PQDINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element*
Real	POREAL*m*		VT_R4	±3.4e38
	PAREAL*m*		VT_R4	±3.4e38
	PQREAL*m*		VT_R4	±3.4e38
Real Array	POREAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element*
	PAREAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element*
	PQREAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element*

*: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

**: For DDE/SuiteLink, this value is restricted to the range of 0 to 2147483647. Values higher than that are clamped to the maximum value of 2147483647 in a SuiteLink or DDE client. In this case, the quality of the item shows "Clamp High."

The S7-200 PLCs do **not** support Peripheral Output Bytes.

Where:

x	is the start address, with a range from 0 to 65535.
y	is the bit position, with a range from 0 to 7. • 0 is the LSB (Least Significant Bit). • 7 is the MSB (Most Significant Bit).
n	is the start address of 2-byte data/2-byte data arrays, with a range from 0 to 65534.
m	is the start address of 4-byte data/4-byte data arrays, with a range from 0 to 65532.
v	is the length of data in bytes, with a range from 1 to (net PDU data size/type size - header information).

Note All peripheral outputs are **Write-Only**. The longest string or array that can be read in a cyclic service is the length of the PDU size minus 32 bytes. The longest string the InTouch software can process is 131 bytes. The longest string that can be poked is 256 bytes or the PDU size minus 28 bytes, whichever is less. All output modules are not readable. Only POKES are allowed.

Counters

The following table summarizes data format, item or point, suffix, data type, and range for Counters.

Data Format	Item/Point	Suffix	Data Type	Range
Word	C*x*	None	VT_UI2	0…65535
	Z*x*	None	VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s

Where:

x is the start address, with a range from 0 to 65535.

When the suffix is used, the client is responsible for ensuring the correct value is specified or returned.

For example:

If the value of counter C0 is 42, the value read by just using the "C0" item syntax is 42. However, if the item syntax is "C0 BCD," the value returned by the DAServer is 66.

Similar principle applies to poking:

If the value for the counter is 42, using the "C0" pokes a value of 42 into the counter C0. However, with the "C0 BCD" syntax, the poke value is 66.

Note All counters are **Read/Write**. The SIDirect DAServer processes a write (**POKE**) to a counter. Although the SIDirect DAServer allows poking any word value into counters, the S7 PLC can only process values in the range of 0…2457 or 0…999 (BCD).

Timers

For S7-200 Timers, see S7-200 Timers on page 88.

The following table summarizes data format, item or point, suffix, data type, and range only for the S7-300 and S7-400 Timers.

Data Format	Item/Point	Suffix	Data Type	Range
Word	T*x*	None	VT_UI2	0 to 14745
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
	TREAL*x*	None	VT_R4	0.0 to 9990.00

Where:

x is the start address, with a range from 0 to 65535.

When the suffix is used, the client is responsible for ensuring the correct value is specified or returned.

For example:

If the value of timer T0 is 42, the value read by just using the "T0" item syntax is 42. However, if the item syntax is "T0 BCD," the value returned by the DAServer is 66.

Similar principle applies to poking:

If the value for the timer is 42, using the "T0" pokes a value of 42 into the timer T0. However, with the "T0 BCD" syntax, the poke value is 66.

Note All timers are **Read/Write**. The SIDirect DAServer processes a write (**POKE**) to a timer. Although the SIDirect DAServer allows poking any word value into timers, the S7 PLC can only process values that represent a valid time format.

Block Items

The SIDirect DAServer supports Block Items for the S7-300 and S7-400 PLCs. The server does **not** support Block Items for the S7-200 PLCs.

The Block Items have two sets of items:

- Read-Only Block Items
- Write-Only Block Items

Read-Only Block Items

The following table summarizes the data format, item or point, suffix, data type, and range for Read-Only Block Items.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	BL*d*,X*x.y*		VT_BOOL	0 or 1
String	BL*d*,S*x,v*		VT_BSTR	String
	BL*d*,STRING*x,v*		VT_BSTR	String
Byte	BL*d*,B*x*		VT_UI1	0 to 255
	BL*d*,BYTE*x*		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Byte Array	BL*d*,B*x,v*		VT_ARRAY}VT_UI1	0 to 255 for each element*
	BL*d*,BYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
Char	BL*d*,CHAR*x*		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Char Array	BL*d*,CHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
Word	BL*d*,W*n*		VT_UI2	0 to 65535
	BL*d*,WORD*n*		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31
Word Array	BL*d*,W*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	BL*d*,WORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*

Data Format	Item/Point	Suffix	Data Type	Range
Integer	BL*d*,INT*n*		VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	BL*d*,INT*n,v*		VT_BSTR	-32768 to 32767 for each element*
Double Word	BL*d*,D*m*		VT_UI4	0 to 4294967295**
	BL*d*,DWORD*m*		VT_UI4	0 to 4294967295**
		BCD	VT_UI4	0 to 99999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Word Array	BL*d*,D*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	BL*d*,DWORD*m,v*			0 to 4294967295 for each element**
Double Integer	BL*d*,DINT*m*		VT_I4	-2147483648 to 2147483647
		BCD	VT_I4	-9999999 to 9999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Integer Array	BL*d*,DINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element
Real	BL*d*,REAL*m*		VT_R4	±3.4e38
Real Array	BL*d*,REAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element*

*: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

**: For DDE/SuiteLink, this value is restricted to the range of 0 to 2147483647. Values higher than that are clamped to the maximum value of 2147483647 in a SuiteLink or DDE client. In this case, the quality of the item shows "Clamp High."

Where:

x	is the start address, with a range from 0 to 65535.
y	is the bit position, with a range from 0 to 7. • 0 is the LSB (Least Significant Bit). • 7 is the MSB (Most Significant Bit).
n	is the start address of 2-byte data/2-byte data arrays, with a range from 0 to 65534.
m	is the start address of 4-byte data/4-byte data arrays, with a range from 0 to 65532.
v	is the length of data in bytes, with a range from 1 to (net PDU data size/type size - header information).

Note All Block items are **Read-Only**. The longest string or array that can be read in a block service is the length of 65534 bytes. The longest string the InTouch software can process is 131 bytes.

Write-Only Block Items

The following table summarizes the data format, item or point, suffix, data type, and range for Write-Only Block Items.

Data Format	Item/Point	Suffix	Data Type	Range
Byte Array	BW*d*		VT_ARRAY|VT_UI1	0 to 255 for each element*,** Note also that theDAServer does **not** cache the value of the item written to the PLC.
	BWC*d-q,x*		VT_ARRAY|VT_UI1	0 to 255 for each element*,*** Note also that the DAServer caches the value of the item written to the PLC.

Data Format	Item/Point	Suffix	Data Type	Range
	BWC*d-q.Send*		VT_BOOL	TRUE (1)***,**** FALSE (0)***,**** The value of item (BWC*d-q.x*), cached by the DAServer, is sent to the PLC when this item is transitioned from FALSE to TRUE. The value of this item remains TRUE until the appropriate acknowledgment is received from the PLC. After the acknowledgment is received from the PLC, the value of this item is set to 0.

*: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

: The starting address is always at 0 for **Writes. The number of bytes written to the PLC block is determined from the length of the input byte stream. However, the length of the byte stream must be less than 65535.

***: The Siemens S7 block write protocol BSEND does not allow a starting address. All block writes to the PLC always start at address 0. To circumvent this limitation, the BWC item syntax allows for a starting address. The logical number of bytes written to the PLC block is determined from the difference in the start address and end address, (q-x+1), or the length of the input byte stream.

If the input byte stream is longer than the (q-x+1), only (q-x+1) bytes are written.

If the input byte stream is shorter than (q-x+1), the whole input byte stream is written.

Internally, the DAServer allocates cache buffers based on the PLC data block and the end address. For more information, see Block Write Caching.

****: The *BWCd-q.Send* item is readable to allow monitoring of the status of the block send function.

Where:

d is the block ID, in decimal, with a range from 0 to 4294967295.

x is the start address, with a range from 0 to 65535.

q is the end address, with a range from 0 to 65535. It must be equal or greater than x.

Note All BW and BWC block items are **Write-Only**. BWCxx.Send items are **Read/Write**. The longest string or array that can be written in a block service is the length of 65534 bytes. The longest string the InTouch software can process is 131 bytes.

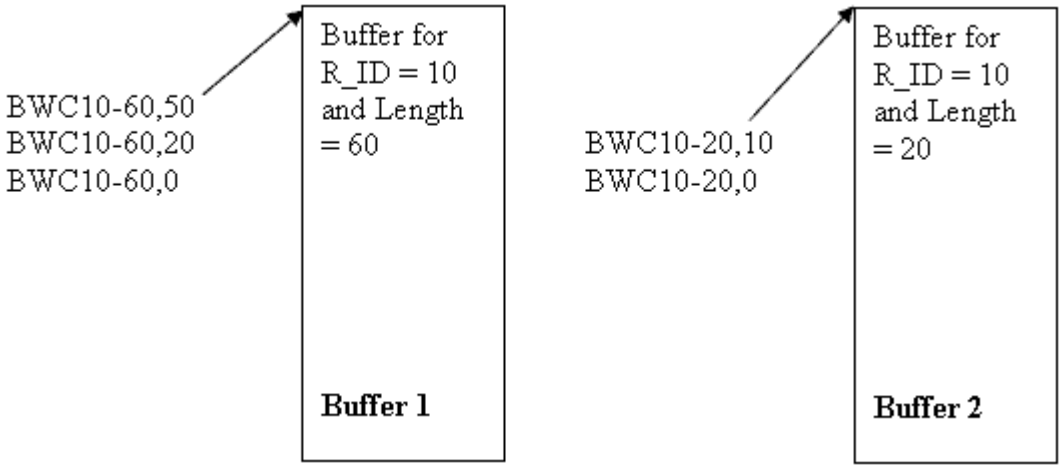
Block Write Caching

BWC is a special item that caches the data before sending it to the PLC. The item syntax for the BWC item is:

BWC<R_ID>-<length>,<start_address>

The R_ID is the ID that is configured for the Block Write in the PLC. A caching buffer is created, based on R_ID and length as an index, inside the server whenever you advise the BWC item. The buffer is not created if one already exists for the same R_ID and length.

For example, you advise the following items: BWC10-60,50, BWC10-60,20, BWC10-60,0, BWC10-20,10 and BWC10-20,0. Only two buffers are created inside the server for these items:



Consider following scenarios:

1 You poke 20 bytes of data in item BWC10-60,0. Buffer-1 is filled from byte 0 to 19.

2 You poke 15 bytes of data in item BWC10-60,50. Buffer-1 is filled from byte 50 to 59. The last 5 bytes are truncated because only 10 bytes are available in that offset.

3 You poke 30 bytes of data in item BWC10-60,20. Buffer-1 is filled from byte 20 to 49.

4 You poke 20 bytes of data in item BWS10-20,0. Buffer-2 is filled from byte 0 to 19.

5 You poke 1 to item BWC10-20.Send. Buffer-2 is flushed to the PLC with byte 20.

6 You poke 1 to item BWC10-60.Send. Buffer-1 is flushed to the PLC with byte 60.

Alarms and Events

Alarm and event information can be received from the S7-300 and S7-400 PLCs. The item syntax for Alarms and Events is as follows:

ALARM<*EV_ID*>.<Extension 1>[,<Extension 2>[<Suffix>]]

EVENT<*EV_ID*>.<Extension 1>[,<Extension 2>[<Suffix>]]

The Alarms and Events table contains valid values and valid value combinations for Extension 1, Extension 2, and Suffix.

Item	Extension 1	Extension 2	Suffix	Data Type	Range
ALARM	EVENT_ STATE			VT_UI2	0 to 65535
<*EV_ID*>	STATE			VT_UI2	0 to 65535
	ACK_STATE			VT_UI2	0 to 65535
	TIME_ STAMP			VT_BSTR	String*, ****
	NO_ADD_ VALUES			VT_UI2	0 to 10
	ADD_VALUE*w*	DATA_TYPE		VT_BSTR	String
		LENGTH		VT_UI2	0 to 65535
		X*x.y*		VT_BOOL	0 or 1
		S*x,v*		VT_BSTR	String
		STRING*x,v*		VT_BSTR	String
		B*x*		VT_UI1	0 to 255
		BYTE*x*		VT_UI1	0 to 255
			DT	VT_BSTR	String
		B*x,v*		VT_ARRAY | VT_UI1	0 to 255 for each element**
		BYTE*x,v*		VT_ARRAY | VT_UI1	0 to 255 for each element**
		CHAR*x*		VT_I1	-128 to 127
			DT	VT_BSTR	String
		CHAR*x,v*		VT_ARRAY | VT_I1	-128 to 127 for each element**
		W*n*		VT_UI2	0 to 65535
		WORD*n*		VT_UI2	0 to 65535
			BCD	VT_UI2	0 to 9999
			KT	VT_BSTR	0.0 to 999.3
			S5T	VT_BSTR	0ms to 2h46m30s
			D	VT_BSTR	String
		W*n,v*		VT_ARRAY | VT_UI2	0 to 65535 for each element**
		WORD*n,v*		VT_ARRAY | VT_UI2	0 to 65535 for each element**
		INT*n*		VT_I2	-32768 to 32767
			BCD	VT_I2	0 to 9999
			D	VT_BSTR	String

Item	Extension 1	Extension 2	Suffix	Data Type	Range
		INT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element**
		D*m*		VT_UI4	0 to 4294967295***
		DWORD*m*		VT_UI4	0 to 4294967295***
			BCD	VT_UI4	0 to 99999999
			T	VT_BSTR	String
			TOD	VT_BSTR	String
		D*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element**
		DWORD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element**
		DINT*m*		VT_I4	-2147483648 to 2147483647
			BCD	VT_I4	0 to 99999999
			T	VT_BSTR	String
			TOD	VT_BSTR	String
		DINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element**
		REAL*m*		VT_R4	±3.4e38
		REAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element**

Item	Extension 1	Extension 2	Suffix	Data Type	Range
EVENT <*EV_ID*>	EVENT_ STATE			VT_UI2	0 to 65535
	STATE			VT_UI2	0 to 65535
	ACK_STATE			VT_UI2	0 to 65535
	TIME_ STAMP			VT_BSTR	String*, ****
	NO_ADD_ VALUES			VT_UI2	0 to 10
	ADD_VALUE*w*	DATA_TYPE		VT_BSTR	String
		LENGTH		VT_UI2	0 to 65535
		X*x.y*		VT_BOOL	0 or 1
		S,*v*		VT_BSTR	String
		STRING*x,v*		VT_BSTR	String
		B*x*		VT_UI1	0 to 255
		BYTE*x*		VT_UI1	0 to 255
			DT	VT_BSTR	String
		B*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element**
		BYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element**
		CHAR*x*		VT_I1	-128 to 127
			DT	VT_BSTR	String
		CHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element**

Item	Extension 1	Extension 2	Suffix	Data Type	Range
		W*n*		VT_UI2	0 to 65535
		WORD*n*		VT_UI2	0 to 65535
			BCD	VT_UI2	0 to 9999
			KT	VT_BSTR	0.0 to 999.3
			S5T	VT_BSTR	0ms to 2h46m30s
			D	VT_BSTR	String
		W*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element**
		WORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element**
		INT*n*		VT_I2	-32768 to 32767
			BCD	VT_I2	0 to 9999
			D	VT_BSTR	String
		INT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element**
		D*m*		VT_UI4	0 to 4294967295***
		DWORD*m*		VT_UI4	0 to 4294967295***
			BCD	VT_UI4	0 to 99999999
			T	VT_BSTR	String
			TOD	VT_BSTR	String
		D*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element**
		DWORD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element**
		DINT*m*		VT_I4	-2147483648 to 2147483647
			BCD	VT_I4	0 to 99999999
			T	VT_BSTR	String
			TOD	VT_BSTR	String
		DINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element**
		REAL*m*		VT_R4	±3.4e38
		REAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element**

*: Starting with version 1.1, this string value is used to timestamp other alarm items.

**: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

***: For DDE/SuiteLink, this value is restricted to the range of 0 to 2147483647. Values higher than that are clamped to the maximum value of 2147483647 in a SuiteLink or DDE client. In this case, the quality of the item shows "Clamp High."

****: For alarm blocks (such as SFB31, 34, 35) that provide multiple event states in one notification, the timestamp that comes with the notification reflects only the timestamp of the last changing state. This restriction is prescribed by the message that it receives from the PLC.

Where:

<EV_ID>	is the ID defined by Step7, in the integer format, filled with leading zeros up to six (6) characters.
x	is the start address, with a range from 0 to 65535.
y	is the bit position, with a range from 0 to 7. • 0 is the LSB (Least Significant Bit). • 7 is the MSB (Most Significant Bit).
n	is the start address of 2-byte data/2-byte data arrays, with a range from 0 to 65534.
m	is the start address of 4-byte data/4-byte data arrays, with a range from 0 to 65532.
v	is the length of data in bytes, with a range from 1 to (net PDU data size/type size - header information).
w	is the length of the net S7 string-data in characters, with a range from 1 to 10.

Note All alarms and events are **Read-Only**. The longest string or array that can be read in a cyclic service is the length of the PDU size minus 32 bytes. The longest string the InTouch software can process is 131 bytes. The DAServer does **not** process writes (**POKES**) to Alarms and Events.

Examples:

ALARM000010.TIME_STAMP

EVENT001234.ADD_VALUE2,LENGTH

ALARM000555.ADD_VALUE10,REAL0

EVENT000001.ADD_VALUE3,D0TOD

Alarm and event information is delivered from the S7-300/400 PLCs in the form of data items with the syntax described above. Alarm provider and alarm acknowledgment functionality is **not** supported.

Note The S7-200 PLCs do **not** provide alarms and events capability.

The configuration of the alarms is first performed in the appropriate function blocks in the S7-300/400 PLCs as follows:

Name	SFB/SFC	S7 CPU
ALARM_SQ	SFC 17	S7-300/400
ALARM_S	SFC 18	S7-300/400
ALARM_DQ	SFC 107	S7-300/400
ALARM_D	SFC 108	S7-300/400
NOTIFY_8P	SFB 31	S7-400
ALARM	SFB 33	S7-400
ALARM_8	SFB 34	S7-400
ALARM_8P	SFB 35	S7-400
NOTIFY	SFB 36	S7-400

Events must be configured in the Symbol Editor.

Alarms and Events Terms

The following table lists the terms available in Alarms and Events Terms and their descriptions.

Term	Description
EVENT_STATE:	State of the Alarm/Event itself. If the Alarm/Event is TRUE, then EVENT_STATE is TRUE and vice versa. For more detailed information, see the Siemens Step7 documentation.
STATE:	The state in general whether the Alarm/Event is available. Maybe a data block is deleted where a bit should be monitored.
ACK_STATE:	The state of the acknowledgment of coming or going Alarms/Events. For more detailed information, see the Siemens Step7 documentation.
TIME_STAMP:	The timestamp of the Alarm/Event provided by the PLC.

Term	Description
NO_ADD_VALUES:	The number of additional values sent with this Alarm/Event message.
ADD_VALUE*w*,DATA_TYPE:	The data type of a specific additional value of an Alarm/Event.
ADD_VALUE*w*,LENGTH:	The length of a specific additional value of an Alarm/Event.
<*EV_ID*>:	The event ID created automatically by the Step7 programming software. In case of Alarms (FB33 to FB36, SFC17/18/107/108), this is the EV_ID-parameter of the function block. The value of the parameter must be converted from hexadecimal to decimal, and then filled up with leading zeros to the length of 6 characters (for example: EV_ID: DW#16#4E25 => <*EV_ID*>: 020005). In case of Events (generated by the symbol editor) this is the "Message number." This number is in decimal format and must be filled with leading zeros up to 6 characters (for example: "Message number": 20000 => <*EV_ID*>: 020000).

Specific S7-200 Item Syntax

The S7-200 PLCs use different syntax than the S7-300 and S7-400 PLCs.

The memory address range in the S7-200-series PLCs varies based on the size of the CPU. Therefore, you can configure the SIDirect DAServer to read and write a PLC memory address that does not exist in the PLC. To prevent this from happening, ensure that the configured PLC item syntax does not exceed the PLC memory ranges of the PLC you are using.

V Memory

The S7-200 PLCs support V Memory.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	VX*x.y*		VT_BOOL	0 or 1
String	VS*x,v*		VT_BSTR	String
	VSTRINGx,v		VT_BSTR	String
Byte	VB*x*		VT_UI1	0 to 255
	VBYTE*x*		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Byte Array	VB*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
	VBYTE*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element*
Char	VCHAR*x*		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999
Char Array	VCHAR*x,v*		VT_ARRAY|VT_I1	-128 to 127 for each element*
Word	VW*n*		VT_UI2	0 to 65535
	VWORDn		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31
Word Array	VW*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	VWORD*n,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*

Data Format	Item/Point	Suffix	Data Type	Range
Integer	VINT*n*		VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	VINT*n,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
Double Word	VD*m*		VT_UI4	0 to 4294967295**
	VDWORD*m*		VT_UI4	0 to 4294967295**
		BCD	VT_UI4	0 to 9999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Word Array	VD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	VDWORD*m,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
Double Integer	VDINT*m*		VT_14	-2147483648 to 2147483647
		BCD	VT_14	-9999999 to 9999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648 MS to 24D_20H_31M_23S_647MS
Double Integer Array	VDINT*m,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element*
Real	VREAL*m*		VT_R4	±3.4e38
Real Array	VREAL*m,v*		VT_ARRAY|VT_R4	±3.4e38 for each element*

*: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

**: For DDE/SuiteLink, this value is restricted to the range of 0 to 2147483647. Values higher than that are clamped to the maximum value of 2147483647 in a SuiteLink or DDE client. In this case, the quality of the item shows "Clamp High."

Where:

x	is the start address, with a range from 0 to 10239.
y	is the bit position, with a range from 0 to 7. • 0 is the LSB (Least Significant Bit). • 7 is the MSB (Most Significant Bit).
n	is the start address of 2-byte data/2-byte data arrays, with a range from 0 to 10238.
m	is the start address of 4-byte data/4-byte data arrays, with a range from 0 to 10236.
v	is the length of data in bytes, with a range from 1 to (net PDU data size/type size - header information).

Note All V Memory bytes are **Read/Write**. The longest string or array that can be read in a cyclic service is the length of the PDU size minus 32 bytes. The longest string the InTouch software can process is 131 bytes. The longest string that can be poked is 256 bytes or the PDU size minus 28 bytes, whichever is less. The SIDirect DAServer processes a write (**POKE**) to V Memory.

Analog Input

The following table summarizes the data format, item or point, suffix, data type, and range for Analog Input.

Data Format	Item/Point	Suffix	Data Type	Range
Word	AIW*x*		VT_UI2	0 to 65535
	AIWORD*x*		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31
Word Array	AIW*x,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	AIWORD*x,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
Integer	AIINT*x*		VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	AIINT*x,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*

*: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

Where:

x is the start address, with a range from 0 to 62.

v is the length of data in bytes, with a range from 1 to (net PDU data size/type size - header information).

Analog Output

The following table summarizes the data format, item or point, suffix, data type, and range for Analog Output.

Data Format	Item/Point	Suffix	Data Type	Range
Word	AOW*x*		VT_UI2	0 to 65535
	AQW*x*			
	AOWORD*x*		VT_UI2	0 to 65535
	AQWORD*x*			
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31
Word Array	AO*x,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	AQ*x,v*			
	AOWORD*x,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	AQWORD*x,v*			
Integer	AOINT*x*		VT_I2	-32768 to 32767
	AQINT*x*			
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	AOINT*x,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
	AQINT*x,v*			

*: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

Where:

x	is the start address, with a range from 0 to 30.
v	is the length of data in bytes, with a range from 1 to (net PDU data size/type size - header information).

Stage Memory

The following table summarizes the data format, item or point, suffix, data type, and range for Stage Memory.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	SX*x.k*		VT_BOOL	0 or 1
Byte	SB*x*		VT_UI1	0 to 255
Byte Array	SB*x,v*		VT_ARRAY|VT_UI1	0 to 255*
Word	SW*y*		VT_UI2	0 to 65535
	SWORD*y*		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31
Word Array	SW*y,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
	SWORD*y,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element*
Integer	SINT*y*		VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	SINT*y,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
Double Word	SD*z*		VT_UI4	0 to 4294967295**
	SDWORD*z*		VT_UI4	0 to 4294967295**
		BCD	VT_UI4	0 to 99999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Word Array	SD*z,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*
	SDWORD*z,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element*

Data Format	Item/Point	Suffix	Data Type	Range
Double Integer	SDINT*z*		VT_I4	-2147483648 to 2147483647
		BCD	VT_I4	-9999999 to 9999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Integer Array	SDINT*z,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element*

*: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

**: For DDE/SuiteLink, this value is restricted to the range of 0 to 2147483647. Values higher than that are clamped to the maximum value of 2147483647 in a SuiteLink or DDE client. In this case, the quality of the item shows "Clamp High."

Where:

k is the bit position, with a range from 0 to 7.
- 0 is the LSB (Least Significant Bit).
- 7 is the MSB (Most Significant Bit).

x is the start address, with a range from 0 to 31.

y is the start address, with a range from 0 to 30.

z is the start address, with a range from 0 to 28.

v is the length of data in bytes, with a range from 1 to (net PDU data size/type size - header information).

Special Memory

The following table summarizes the data format, item or point, suffix, data type, and range for Special Memory.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	SM*x.k*		VT_BOOL	0 or 1
Byte	SMB*x*		VT_UI1	0 to 255
Byte Array	SMB*x,v*		VT_ARRAY|VT_UI1	0 to 255 for each element**
Word	SMW*y*		VT_UI2	0 to 65535
	SMWORD*y*		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31
Word Array	SMW*y,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element**
	SMWORD*y,v*		VT_ARRAY|VT_UI2	0 to 65535 for each element**
Integer	SMINT*y*		VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	SMINT*y,v*		VT_ARRAY|VT_I2	-32768 to 32767 for each element*
Double Word	SMD*z*		VT_UI4	0 to 4294967295***
	SMDWORD*z*		VT_UI4	0 to 4294967295***
		BCD	VT_UI4	0 to 99999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Word Array	SMD*z,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element**
	SMD*WORDz,v*		VT_ARRAY|VT_UI4	0 to 4294967295 for each element**

Data Format	Item/Point	Suffix	Data Type	Range
Double Integer	SMDINT*z*		VT_I4	-2147483648 to 2147483647
		BCD	VT_I4	-9999999 to 9999999
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Integer Array	SMDINT*z,v*		VT_ARRAY|VT_I4	-2147483648 to 2147483647 for each element

Special Memory addresses from 0 to 29 are **Read-Only**. Addresses above 29 are both **Read** and **Write**. If you perform a poke (a Write request) using addresses 0 through 29, the PLC will reject it.

**: For DDE/SuiteLink, the item value is the HexASCII representation of the complete array. The result is one string containing all the elements of the array in the HexASCII representation of the binary data in big-endian format when data is returned to the DDE/SuiteLink layer.

***: For DDE/SuiteLink, this value is restricted to the range of 0 to 2147483647. Values higher than that are clamped to the maximum value of 2147483647 in a SuiteLink or DDE client. In this case, the quality of the item shows "Clamp High."

Where:

k is the bit position, with a range from 0 to 7.
- 0 is the LSB (Least Significant Bit).
- 7 is the MSB (Most Significant Bit).

x is the start address, with a range from 0 to 549.

y is the start address, with a range from 0 to 548.

z is the start address, with a range from 0 to 546.

v is the length of data in bytes, with a range from 1 to (net PDU data size/type size - header information).

High-Speed Counter

The following table summarizes the data format, item or point, suffix, data type, and range for High-Speed Counter.

High speed counters are restricted to items HC0 through HC5. The DAServer returns Bad quality to high speed counter items with an address higher than 5.

The following table summarizes the data format, item or point, suffix, data type, and range for High-Speed Counter.

Data Format	Item/Point	Suffix	Data Type	Range
Double Word	HC*x*		VT_I4	-2147483648 to 2147483647*

*: The high speed counter item is **Read-Only**. The item type is double word. The value ranges only from -2147483648 to 2147483647.

Where:

x is the start address, with a range from 0 to 5.

S7-200 Timers

The following table summarizes the data format, item or point, suffix, data type, and range for S7-200 Timers.

Data Format	Item/Point	Suffix	Data Type	Range
Word	T*x*		VT_UI2	0 to 32767*,***
Double Word	T*x*	MS	VT_UI4	0 to 3276700 milliseconds**,***

*: The T*x* item syntax returns or pokes the timer value in its native form as stored in the PLC. The timer resolution may be different for different timer addresses. Be aware of the timer resolution and convert it to the proper value for processing. See *** below.

: The T*x* MS item syntax translates the S7-200 counter value to its corresponding time in milliseconds. The SIDirect DAServer performs the conversion automatically. It is based on the timer address, x, specified in the item syntax using the timer resolution table stated in the * note.

***: The timer resolution is based on its address listed in the following table.

Timer Address	Resolution in Milliseconds	Maximum Value in Milliseconds
T0, T32, T64, T96	1	32767
T1-T4, T33-T36, T65-T68, T97-T100	10	327670
T5-T31, T37-T63, T69-T95, T101-T255	100	3276700

For example, if the PLC's T33 timer value is 1000, advising item T33 returns a timer value of 1000, advising item T33 MS returns a value of 10000 milliseconds. Similarly, if T255 timer in the PLC has a value of 2000, advising item T255 MS returns a value of 200000 milliseconds. Poking a value of 5000 to T33 MS writes a value of 500 to the T33 timer in the S7-200 PLC.

Where:

x is the start address, with a range from 0 to 255.

Note All timers are **Read** and **Write**.

Conversions and Suffixes of Items

This section describes what data-format items and suffixes are converted and what they are converted into.

Endian Conversion

In endian conversions, all items with the following data formats are copied in a reverse-byte order to convert the data from the big endian of the PLC to the little endian of the computer:

- Word
- Integer
- Double Word
- Double Integer
- Real

Suffix BCD

All items with the following data formats and suffix BCD are converted from the BCD format into the integer and back:

- Word
- Integer
- Double Word
- Double Integer

Suffix DT

All items with the following data formats and suffix DT (Date and Time) are converted from DT into a message and back to store a value in the range of 1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999."

- Byte
- Char

This is an 8-byte value (although declared as "byte") that contains both the date and time. In the client, you see a string such as: 1999-12-13-07:06:05.888. The construction is a BCD interpretation. This means that the value in the memory of the PLC (seen as a hex value) represents directly the single "parts" of the string above.

The example above looks like the following in the memory:

0x9912130706058880

The last character ("0" in this example) is not used in this string, but represents the day of the week. If a DT item is poked, the server writes the correct day of the week to the PLC.

Suffix KT

All items with the following data format and suffix KT are converted from KT to a message and back to store a value in the range of 0.0 to 999.3.

- Word

The item contains a time value in the same format as in the old Step-5 PLCs. In the client, you see a string such as: 999.3. The construction is like a BCD interpretation, but the digits are twisted.

The example above looks like the following in the memory of the PLC:

0x3999

Another example, 0x2345, in the memory of the PLC is 345.2 as the item value.

Suffix S5T

All items with the following data format and suffix S5T are converted from S5T to a message and back to store a value in the range of 0ms to 2h46m30s.

- Word

The memory in the PLC is exactly the same as for the KT items, but the presentation is different although it means the same time. This means a memory content of 0x3999 (as in the first example for KT) results in the string of 2h46m30s0ms.

999.3 (KT) means as follows:

999	The first three characters are the time value in BCD.
3	The last digit is the multiplier. Possible values are: 0: 0.01s 1: 0.1s 2: 1s 3: 10s.

This means:

A value of 123.0 represents:	123*0.01s=1.23s (equals 1s230ms)
A value of 543.2 represents:	543*1s=543s (equals 9m3s0ms)
A value of 999.3 represents:	999*10s=9990s (equals 2h46m30s0ms)

Suffix TR

All items with the following data format and with suffix TR (Time as real value) are converted from TR into a real value or back to store a value in the range of 0.0 to 9990.0 (s).

- Word

The memory in the PLC is exactly the same as for the KT items, but the presentation is different, although it means the same time. The memory content of 0x3999 (as in the example for KT) results in the real value of 9990.0. The construction is the result of the multiplication as described in the examples for S5T, given to the client as a real value.

Suffix D

All items with the following data formats and with suffix D (Date) are converted from D into a message or back to store a value in the range of 1990-1-1 to 2168-12-31.

- Word
- Integer

The item contains the date. The construction is the number of days since 1/1/1990. The integer value 4010, for example, represents 2000-12-24.

Suffix T

All items with the following data formats and with suffix T (Time) are converted from T into a message or back to store a value in the range of –24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS.

- Double Word
- Double Integer

The item contains a time in the IEC format. The client shows a value such as: 3D_7H_32M_11S_153MS. This is the time in milliseconds, shown as a more readable string.
The range from 0 to 2147483647 (0x0 to 0x7FFFFFFF) is interpreted as a positive time value. The range from -2147483648 to -1 (0x80000000 to 0xFFFFFFFF) is interpreted as a negative time value.

Suffix TOD

All items with the following data formats and with suffix TOD (Time Of Day) are converted from TOD into a message or back to store a value in the range of 0:00:00.000 to 23:59:59.999.

- Double Word
- Double Integer

The item contains the time of a day. The client shows a value such as: 4:58:15.654. This is the time in milliseconds (as for T), shown as a more readable string. The highest value is 23:59:59.999. There are no negative values. All values greater than 86399999 (0x05265BFF) are shown with quality 0x0056 (Clamp Hi).

Note If you use the DAServer in the English operating system, the following applies. The string is always represented in a 24-hour format, regardless of the time representation of the operating system used. This means you see the time 1:13:5 P.M. as 13:13:5.0.

DAServer Standard System Items

System items supply you with easy access to the DAServer status and diagnostic 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 tracking of the DAS Engine.

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

System items, like ordinary items, are defined by name in the following context:

- **Group** (client group/OPC group)
  Arbitrary collection of items, not correlated.
- **Hierarchical location** (link name/OPC path:
  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)
  Collection of items on the same physical location with the same protocol update rate.

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

```
<PortCpS7 name>.<S7Cp>.$SYS$Status
```

Example:

```
TCPIP.PLC1.$SYS$Status
```

In this example, the scope of the item is not limited to a specific access path/device group. As long as the data requested is from the same external device specified by the same hierarchical location, the value is always the same.

Note For DDE/SuiteLink clients, SYSStatus always comes from the leaf level of a DAServer hierarchy branch, which is defined by the unique device group. For OPC clients, SYSStatus can be accessed at all hierarchy levels. SYSStatus 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 SYSStatus at 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.
- Parsing of the name is case-insensitive.
  The DAS Engine scans and parses the name for system items.

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
SYSLicensed	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 licenses exist.

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
SYSStatus	Boolean/ Read	The 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. The status can be good even if individual items have errors. For DDE/SuiteLink clients, SYSStatus always comes from the leaf level of a DAServer hierarchy branch, which is the destination PLC node. For OPC clients, SYSStatus can be accessed at all hierarchy levels. SYSStatus 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 SYSStatus at any hierarchy levels other than the root.	RANGE: 0, 1 1: DAServer connection to the device is intact. 0: Error communicating with the device.
SYSErrorCode	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 such as, connecting, initializing, and so on. <0: Error status (value indicates the error).

System Item Name (Type)	Type/ Access Rights	Description	Values
SYSErrorText	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.
SYSStoreSettings	Integer/Read/Write	Makes the temporary update interval changes via the SYSUpdateInterval item permanent. If the client pokes a value of 1 into this system item, the currently set update interval is written to the servers 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 is changed via the SYSUpdateInterval 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 you set 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: Always Read value 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 are configured in the DAServer.

System Item Name (Type)	Type/ Access Rights	Description	Values
SYSUpdateInterval	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 no non-system items on that topic are updated (data for these items are not acquired from the device).	RANGE: 0…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.
SYSMaxInterval	Dword/Read	Accesses 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 shown.	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
SYSWriteComplete	Integer/Read/Write	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. 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 completes all write activities, the value of this item changes to 1 if all pokes were successful or to -1 if at least one poke 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.

System Item Name (Type)	Type/ Access Rights	Description	Values
SYSReadComplete	Integer/Read/Write	Accesses the state of reads on all items in the corresponding device group. The value is unequal 0 if all active items in a device group are read. If at least one item in the device group is activated, this item changes to 0. It changes to 1 if all newly activated items are 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.
SYSItemCount	Dword/Read	Accesses the number of items in the corresponding device group. This item is read-only.	RANGE: 0…2147483647 >=0: Number of active and inactive items.
SYSActiveItemCount	Dword/Read	Accesses 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
SYSErrorCount	Dword/Read	Accesses the number of all items (active and inactive) with 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.
SYSPollNow	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 getting 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 is a container for OPC Groups, providing 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. All access to OPC items is maintained through the OPC Group.

The fully qualified name for an OPC item is 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:

```
TCPIP.PLC1.DB1,B20
```

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

In this example:

- PLC1 is the name of the target PLC.
- DB1,B20 is the specific data point or 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 data point.
- OPC provides another parameter, called Access Path, that defines optional specifications for obtaining that data.

In DAServers, Access Paths are equivalent to Device Groups. This parameter defines the update interval between the DAServer and the field device for accessing the values of data points in the PLC.

Chapter 4

Troubleshooting

This section describes the troubleshooting tools that you can use to deal with SIDirect DAServer problems.

The DAServer Manager provides access to diagnostics and other statistical data. The Log Viewer provides access to event messages logged during the operation of the SIDirect DAServer. Your client, for example, the InTouch software, can also monitor connectivity with the PLC through the SYSStatus item. Use these tools together with the information in this chapter to troubleshoot your SIDirect DAServer.

Finding the DAServer Version Number

This section describes finding the version number of your DAServer.

To find the version number

1. From the **Start** menu, point to **Settings**, and click on the **Control Panel** option.
2. Click **Add or Remove Programs**
3. Find and click on **Wonderware SIDirect DAServer** from the **Currently installed programs** list and click the hyperlink, **Click here for support information**. The release version of the DAServer appears in the **Support Info** dialog box.

OR

- Click on the DAServer node in the hierarchy-tree view. In the **Details** pane on the right you see the build version numbers of the respective DAServer components.

OR

1. Search for **DAS**SIDirect**.dll**.
2. Right-click on the **File Name** and select **Properties** on the menu.
3. Click the **Version** tab on the **Properties** dialog box. The version of your DAServer is listed under **File Version**.

Monitoring Connectivity Status with the PLC

You can use the built-in discrete item, SYSStatus, to monitor the status of communications with the PLC. This item is set to:

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

For DDE/SuiteLink clients, SYSStatus always comes from the leaf level of a DAServer hierarchy branch, which is the destination PLC node. For OPC clients, SYSStatus can be accessed at all hierarchy levels. SYSStatus 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 SYSStatus at any hierarchy levels other than the root.

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

=DASSIDirect|S7PLC!SYSStatus

where:

DASSIDirect	is the name of the DAServer application.
S7PLC	is the exact device group defined in the DAServer for the PLC.
SYSStatus	is the discrete item that monitors the status of connectivity with the PLC.

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

YourOPCAccessPath.SYSStatus

where:

YourOPCAccessPath	is the assembly of hierarchy node names leading to a specific device (controller).
SYSStatus	is the discrete item used to monitor the status of connectivity with the device (controller).

Monitoring the Status of DAS Conversations

The **InTouch WindowsViewer** supports built-in topic names, **DDEStatus** and **IOStatus**, that can monitor the status of specific DAS conversations.

For example, assume that **WindowViewer** (**VIEW**) is communicating with the SIDirect DAServer to a PLC. The PLC is defined in the DAServer with the topic name **S7PLC**. 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 software can be read into Excel by entering the following DDE reference formula in a cell on a spreadsheet:

=view|DDEStatus!S7PLC

or

=view|IOStatus!S7PLC

where:

view	is the name of the InTouch application.
[DDE][IO] Status	is the built-in topic name that monitors the status of communications between the DAServer and the InTouch software.
S7PLC	is the exact topic name defined in the server for the PLC.

Reading Values from the DAServer into Excel

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

=applicationname|<devicegroup>!itemname

Example formula:

=DASSIDirect|S7PLC!'DB1,B20'

where:

DASSIDirect	is the name of the DAServer application.

S7PLC	is the exact device group name defined in the DAServer for the PLC.
DB1,B20	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 **<DB1,B20>** changes in the PLC, the DAServer automatically sends the new value to the cell containing the formula in Excel.

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

Writing Values to the DAServer from Excel

You can write values from Microsoft Excel to the DAServer 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 that is defined in the DAServer in a particular application name (the executable name without the .exe).
- Assigns the number of that opened channel to **channel**.

Note When 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 or the actual location in the PLC, via the **channel** number that is 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.

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

=RETURN()

- Marks the end of the macro.

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

Debugging Communications Between the SIDirect DAServer and the PLC

The DAServer Manager lets you use on-line diagnostics of the SIDirect DAServer components at run-time, locally and remotely.

To perform on-line diagnostics

◆ Select any active SIDirect DAServer on any node in the DAServer Manager.

The Diagnostics branch is visible only if the DAServer is active as indicated by the green icon on the server branch. It contains the following sub-branches:

- Client Groups
- Structure
- Transactions
- Statistics
- Messages
- Device Groups

Each of these sub-branches contains live information from the DAServer. They allow detailed diagnostics of objects within the SIDirect DAServer.

Note If you have pokes that are folded, the diagnostics shows ALL items in the transaction. However, because they are folded, only items that have actually been sent have both the message ID and value. All other items that have not been sent, because of the folding, are listed in this transaction with the same timestamp but without the message and value.

Client Groups

The information provided in the Client Groups diagnostic root is organized like all DAServers. See the DAServer Manager user's guide for detailed explanation of the generic diagnostic information of DAServers.

In the Diagnostics Client Groups branch, the OPC client groups are listed as created by the OPC clients. The DDE/SL plug-in always creates only one client group called DDESLPlugIn.

The list view shows the following information:

Name	The name of the client group.
Items	The number of created items in the client group.
Active Items	The number of active items in the client group.
Errors	The number of active and inactive items with errors or the OPC quality is not good in the client group.
Update Interval	The client group update interval.
State	The state of the client group, Active or Inactive.

Clicking an item in the list view shows the full diagnostic information for this item.

For example:

Name:	DDESLPlugIn
Items:	10234
Active Items:	10234
Items with Errors:	0
Update Interval:	0

Selecting a client group in the tree view shows all items in this client group. The item information is grouped into the following columns:

Name	The OPC leaf item name without the fully qualified OPC item ID path.
Client Value	The last updated value to the client.
Client Time	The timestamp the client was updated last.
Client Quality	The item quality of last update.
Subscr Msg	The DAS identifier for the subscription message of the item.

Location	The path item name, which is the OPC path part of the fully qualified item ID.
Device Group	The name of the device group the item is in.

Clicking an item in the list view shows the full diagnostic information of this item.

For example:

Name:	MB9
Client Value:	129
Client Time:	Monday, July 29, 2002
Quality:	GOOD: Nonspecific: …
Location:	New_PortCpS7_000 New_ …
Subscription Message:	1002809
Device Group:	PLC1

Structure

The structure view shows the physical hierarchical organization of the DAServer. In the Diagnostics Structure branch, the structure view shows generic DAServer information as well as SIDirect-DAServer-specific information. See the DAServer Manager user's guide for detailed explanation of the generic diagnostic information of DAServers.

The list view shows either the branches, items, or both, populating different columns. The branches populate the following columns:

Name	The branch name.
Items	The number of active items.
Errors	The number of items with errors.
R/W Status	The Read/Write status of the corresponding branch. R: Read complete All item values are acquired. W: Write complete All write operations are completed.
Messages	The number of messages and messages on this branch.
Device Groups	The number of device groups on this branch.

The SIDirect DAServer provides the following additional S7-specific columns for branches:

S7 Error:	The S7 error code.
A&E:	The Alarms and Events setting. Valid values are OFF, ALARMS ON, and EVENTS ON.

The items populate the following columns:

Name	The item leaf name.
R/W Status	R: Item was updated. W: No writes are pending on the item.
Value	The last protocol value.
Time	The last protocol time.
Quality	The last protocol quality.

No S7-specific columns are populated for items.

Transactions

The transactions view shows currently pending transactions and completed transactions, depending on the backlog setting. It also shows generic DAServer information as well as SIDirect-DAServer-specific information. Refer to the DAServer Manager user's guide for detailed explanation of the generic diagnostic information pertaining to the DAServers.

In the Diagnostics Transactions branch, the transactions populate the following columns:

Type	The Demand Read and Demand Write icons showing the state of the transaction.
ID	The unique identifier for the transaction.
Items	The number of items.
Status	The status of the transaction.
Start	The time the transaction starts.
End	The time the transaction ends. The column is empty if the transaction is not complete.

Expanding a transaction lists all items with the same columns as in Structure.

Statistics

The statistics view shows current statistics of a DAServer. This generic information is not specific to the SIDirect DAServer. See the DAServer Manager user's guide for detailed explanation of the generic diagnostic information pertaining to DAServers.

Messages

The message view displays generic and S7-specific information of the SIDirect DAServer. See the DAServer Manager user's guide for detailed explanation of the generic diagnostic information of DAServers.

In the Diagnostics Messages branch, the message view shows all messages in the DAS Engine.

MsgID	The message unique identifier.
Items	The number of items.
Errors	The number of items with non-good quality.
Status	The status of the message.

The following is the SIDirect-DAServer-specific diagnostic information:

S7 Msg ID	The S7 message ID consisting of one of the following letters: P: poll message C: cyclic service B: block message W: write message S: scan message A: alarm message and a number
PDU size	<req msg>/<data resp> Where <data resp> is the size of the data in the response message and <req msg> is the size of the request message.
Message error	The S7-message communications error code.

Clicking a message in the list view shows the full diagnostic information of this message.

For example:

Name:	0100280A
Active Items:	462

Error Items:	0
Status:	IDLE
S7 Msg ID:	P0
PDU Size (resp/req):	480/24
Msg Error:	OK

Expanding a message lists all items with the same columns as in Structure.

Name	The item leaf name.
R/W Status	R means the item was updated. W means no writes are pending on the item.
Value	The last updated value from the protocol.
Time	The last updated time from the protocol.
Quality	The last updated quality from the protocol.
Msg ID	The DAServer internal message identifier.
Location	The OPC path of fully qualified item ID.

The following is the SIDirect-DAServer-specific diagnostic information:

S7 Msg ID	The S7 message ID consisting of one of the following letters: P: poll message C: cyclic service B: block message W: write message S: scan message A: alarm message and a number
Item error	The S7-item communications error code where 255 = OK.

Clicking an item in the list view shows the full diagnostic information of this item.

For example:

Name:	New_PortCpS7_000.New_...
Read status:	complete
Write status:	complete
Value:	127
Type:	VT_Ul1 - unsigned c...

Time:	Monday, July 29, 2002...
Quality:	GOOD: Nonspecific:...
S7 Msg ID:	P2
Item Error:	OK

Device Groups

The device groups view shows all device groups in the DAS Engine.

Device Group	The device group name.
Update Interval	The protocol update interval in milliseconds.
Items	The number of items.
Active Items	The number of active items.
Errors	The number of items with errors.
Location	The OPC path of the fully qualified Item IDs of items in this device group.
Poll Msgs	The number of poll messages.
Cyclic Srvs	The number of cyclic services.

Clicking a device group in the list view shows the full diagnostic information of this device group.

For example:

Name:	PLC1
Updated Interval:	1000
Number of Items:	10234
Located at:	New_PortCpS7_000.New_...
Poll Messages:	23
Cyclic Services:	0

Expanding a device group in the tree view shows all items in this group. The diagnostic information on items in a device group is identical to the items in Messages.

Diagnostics and Error Tracing

The SIDirect DAServer uses the standard diagnostic information provided by the DAS Toolkit. Access to other internal diagnostic registers of the PLC is performed through reads and writes via the syntax used in Item Naming.

Diagnostics Facility

This section describes the areas in which the SIDirect DAServer diagnostics can help you.

Communications Processor

The diagnostics window of the CP (Communications Processor) shows all DAS-provided diagnostic columns plus a column for any error code on the CP level. The diagnostic information shows DAS information text plus the error text description.

S7 Communications Processor

The diagnostics window of the CP shows all DAS-provided diagnostic columns plus a column for any error code on the PLC CP level. The diagnostic information shows the DAS information text plus the error code description.

Items

To the DAS-provided diagnostics of items, the following two columns are added:

- **The message ID**
  The message ID consists of two parts:
  - A letter indicating the type of message.
    For example, P-poll, C-cyclic, B-block, and so on.
  - A message number.
    For example, order ID, block ID, event ID, and so on.
- **The S7 error code**
  The diagnostic information shows the message ID. For example, order ID, block ID, or event ID, and the error code description.

Messages

Four types of messages having different diagnostic displays.

Variable Service Message with:

- P – poll message
- W – poke message
- C – cyclic message
- B – block message

In addition to the standard DAS diagnostic messages, the following information is added:

- Message ID
- Two PDU sizes indicating block size if applicable and request block size
- The message or S7 error code

The diagnostic information shows the DAS information text plus the message ID, PDU size, and error text description.

Device Group

The device group diagnostics window shows the DAS-provided standard diagnostic columns plus the following four additional columns:

- Number of poll messages
- Number of cyclic services
- Number of block messages
- Number of alarm & event messages

The diagnostic information shows the corresponding additional values.

DASTrace Diagnostic Messages

The SIDirect DAServer generates messages that you can use for diagnostic purposes. These DASTrace diagnostic messages do not necessarily indicate that error conditions exist.

Note The logger messages use the following codes: %s to represent strings, %d to represent numbers (integer), %x to represent the address of the object, and %ums to represent time in milliseconds.

Logger Message	Explanation	Probable Cause	Solution
Message is in SlowPollMode (%s msg=0x%08X) for topic %s	The message state has been switched to Slow poll interval.	The response for this message has not been received by the server. Either the response time is very small or the PLC is very busy.	Adjust the message reply timeout in the server in accordance with the performance of the PLC to achieve the optimal behavior.
Message leaves the SlowPollMode (%s msg=0x%08X) for topic %s	The message state is coming out of the Slow poll interval.	The server recovered from the error situation.	N/A
UpdateInterval for Device Group: %s changed to %d	The update interval for the device group has been changed. The messages are now to be polled at a new update interval for this device group.	N/A	N/A
Setting all items to bad in hierarchy: %s	The server updates the status of all the items as Bad on this connection.	The connection with the PLC is broken.	Check the connection with the PLC.
Connection with PLC failed, restarting connection	The socket connection with the PLC fails due to a given error code.	The PLC is not responding to the Connect request from the DAServer.	Check the error code and take the appropriate action to correct the problem.
Block receive size: doesn't match msg size	The server receives an invalid block packet from the PLC.	This message from the PLC is garbled.	Check the PLC health status and the network condition.

Error Tracing with the Wonderware Logger

The SIDirect DAServer supports error messages, board-specific error messages, trace logger messages, and error codes. Use the Log Flag data to customize the type of messages logged to the Log Viewer.

Note See the Log Viewer online documentation for more information about using log flags.

SIDirect DAServer Logger Flags

The SIDirect DAServer supports the following server-specific DAS logger flags.

- Errors
  General errors from the server have the prefix "ERROR." All related errors, such as for poll messages, have the prefix "POLL_ERROR." They include CONNECTION_ERROR, POLL_ERROR, POKE_ERROR, CYCLIC_ERROR, BLOCK_ERROR, and ALARMS_AND_EVENTS_ERROR.
- Trace
  General traces from the server have the prefix "TRACE." All related traces, such as for poll messages, have the prefix "POLL_TRACE." They include CONNECTION_TRACE, POLL_TRACE, POKE_TRACE, CYCLIC_TRACE, BLOCK_TRACE, and ALARMS_AND_EVENTS_TRACE.

The following table lists all the available logger flags and their meanings.

Type	Name	Description
Server Flag	ERROR	Shows general server errors.
	TRACE	Shows general server traces.

Type	Name	Description
Transaction Flag	CONNECTION_ERROR	Shows connection errors.
	POLL_ERROR	Shows errors of poll messages.
	POKE_ERROR	Shows errors of poke messages.
	CYCLIC_ERROR	Shows errors of cyclic messages.
	BLOCK_ERROR	Shows errors of block messages.
	ALARMS_AND_EVENTS_ERROR	Shows errors of alarm and scan messages.
	CONNECTION_TRACE	Shows connection traces.
	POLL_TRACE	Shows traces of poll messages.
	POKE_TRACE	Shows traces of poke messages.
	CYCLIC_TRACE	Shows traces of cyclic messages.
	BLOCK_TRACE	Shows traces of block messages.
	ALARMS_AND_EVENTS_TRACE	Shows traces of alarm and scan messages.

Error Messages, Trace Messages, Error Codes, and Warnings

In addition to the SIDirect DAServer error and warning messages, S7 Trace messages and generic DAServer error codes are supported. Use these messages together with the DAServer Manager Diagnostic root data to troubleshoot SIDirect DAServer problems.

Note The logger messages use the following codes: %s to represent strings, %d to represent numbers (integer), %x to represent the address of the object, and %ums to represent time in milliseconds.

You can also use the Log Flag data to customize the type of messages logged to the Log Viewer. For more information about using log flags, see the Log Viewer online Help.

DAServer Error Messages

The following list contains error messages produced by the DAServer that are logged to the Log Viewer with the DASProtFail log flags.

Logger Message	Explanation	Probable Cause	Solution
DASProtFail Messages			
Internal Error: CIoVariant::Update()	The internal type conversion encounters an invalid or unknown type.	This is an internal program error.	Turn on POLL_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.
Send fail because of wrong order id	The message cannot be sent to the PLC because of the order ID is incorrect.	This is an internal program error.	Turn on POLL_TRACE, POKE_TRACE, and CYCLIC_TRACE in the Wonderware Logger to obtain additional trace information. If the problem persists, report it to Wonderware Technical Support.
(2):s7_multiple_read_req (orderid=%d) [(%d) %s]x	The poll message cannot be sent to the PLC.	This is an internal program error.	Turn on POLL_TRACE in the Wonderware Logger to obtain additional trace information. If the problem persists, report it to Wonderware Technical support.
Internal state error: multiple read response (S7Type 0x%X)	The server receives a duplicate response for the poll message.	This is an unknown error.	Turn on POLL_TRACE in the Wonderware Logger to obtain additional trace information. If the problem persists, check with the PLC vendor.

Logger Message	Explanation	Probable Cause	Solution
(%d):Invalid item name: %s (%s)	The requested item name has a bad syntax.	The item syntax is wrong.	Correct the item syntax as defined in this user's guide.
(%d):Invalid item name suffix: %s (%s)	The requested item suffix has a bad syntax.	The item suffix is wrong.	Correct the item suffix as defined in this user's guide.
Alarm event header key not: FF09: %02X%02X	The alarm received from the PLC has a bad header.	This is a PLC issue.	Turn on ALARMS_AND_EVENTS_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.
Send: (MSG_FAIL): message (%s msg=0x%08X) [msg_state=%d,con_state=%d]	The server fails to send the message to the PLC.	This is an internal error.	Turn on POLL_TRACE, POKE_TRACE, and CYCLIC_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.
(%d): s7_brcv_init (r_id=%d) [0]	There is an error in s7_brcv_init (Block services).	A communications/ configuration error occured.	Turn on BLOCK_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.
Timeout for initial values of block with r_id=%d	The block message has timed out for the initial updates.	The PLC does not send the block update.	Check the PLC program and see the B_SEND is configured correctly.
Timeout updating values of block with r_id=%d	Timeout occurs while updating the subsequent values for the block message.	The connection with the PLC may be dropped.	Check the PLC program and see the B_SEND is configured correctly.

Logger Message	Explanation	Probable Cause	Solution
Release blockid:%d for message (%s msg=0x%08X,con=%s) [number of blockids=%d] was not successful	Releasing the block ID from the block message operation fails.	This is an internal error.	Turn on BLOCK_TRACE in the Wonderware Logger to obtain additional trace information. If the error persists, report it to Wonderware Technical Support.
S7BlockMessage::HandleResponse: premature block end received at size %d (expected: %d)	The message received from the PLC for the block service has no data.	This is an unknown error.	Turn on BLOCK_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.
S7BlockMessage::HandleResponse: unsegmented block messages did not have correct size: %d (len: %d)	The unsegmented message received from the PLC for the block service has incorrect data size.	This is an unknown error.	Turn on BLOCK_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.
S7BlockMessage::HandleResponse: segmented messages did not add up correctly in size: %d to indicated response data (len: %d)	The segmented message received from the PLC for the block service has incorrect data size.	This is an unknown error.	Turn on BLOCK_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.
S7BlockMessage::HandleResponse: cannot allocate memory for response data (len: %d)	The server cannot allocate the memory for the response received from the PLC for the block message.	The server runs out of memory.	Turn on BLOCK_TRACE in the Wonderware Logger to obtain additional trace information. Restart the server or the computer.

Logger Message	Explanation	Probable Cause	Solution
S7BlockMessage::HandleResponse: too much block data received: (received: %d + new: %d more than expected: %d)	The server receives too much data for the block message.	This is a PLC problem.	Turn on BLOCK_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.
Internal Error: Unknown type: %d updating event item: %s	The server receives the updates for the unknown type of alarm and event message.	This is an unknown error.	Turn on ALARMS_AND_EVENTS_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.
Could not generate data for item %s	The server cannot read the poke value for the item.	This is an internal error.	Turn on POKE_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.
Can't send request for msg=%.08X (no orderid available)	The server cannot send the message to the PLC because it runs out of order IDs.	This is an unknown error.	Turn on POLL_TRACE, POKE_TRACE, and CYCLIC_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.

Logger Message	Explanation	Probable Cause	Solution
Send fail because of wrong order id	The server cannot send the message to the PLC because the order ID is incorrect.	This is an unknown error.	Turn on POLL_TRACE, POKE_TRACE, and CYCLIC_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.
(%d):s7_multiple_write_req (orderid=%d) [0]	The write request to the PLC fails.	The data poke may be too long or this is an internal error.	Turn on POKE_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.
Internal state error: multiple write response (S7Type 0x%X)	The server receives a duplicate response for the poke message.	This is an unknown error.	Turn on POKE_TRACE in the Wonderware Logger to obtain additional trace information. If the problem persists, report the error to Wonderware Technical Support.
(%d):s7_cycl_read (orderid=%d) [0]	An error occurs when sending the S7 cyclic read message.	A communications/ configuration error occured.	Check the PLC configuration/connection and resource limitation, particularly the ones related to the cyclic service.
(%d):s7_cycl_read_init_req (orderid=%d) [0]	An error occurs when sending the S7 cyclic read initiation request message.	A communications/ configuration error occured.	Check the PLC configuration/connection and resource limitation, particularly the ones related to the cyclic service.

Logger Message	Explanation	Probable Cause	Solution
(%d):s7_cycl_read_start_req (orderid=%d): [0]	An error occurs when sending the S7 cyclic read start request message.	A communications/ configuration error occured.	Check the PLC configuration or connection.
(%d):s7_cycl_read_delete_req (orderid=%d): [0]	An error occurs when sending the S7 cyclic read delete message.	A communications/ configuration error occured.	Check the PLC configuration or connection.
ERROR in S7_CYLC_READ_INIT_CNF - PLC responded with error code	The PLC returns an error code in response to the cyclic read initiation request.	It is a PLC, DAServer configuration, or resource error.	Turn on CYCLIC_TRACE in the Wonderware Logger to obtain additional trace information. Check the PLC configuration/connection and resource limitation, particularly the ones related to the cyclic service.
ERROR in S7_CYCL_READ_START_CNF - PLC responded with error code	The PLC returns an error code in response to the cyclic read start request.	It is a PLC or DAServer configuration error.	Turn on CYCLIC_TRACE in the Wonderware Logger to obtain additional trace information. If the problem persists, contact Wonderware Technical Support.
ERROR in S7_CYCL_READ_DELETE_CNF - PLC responded with error code	The PLC returns an error code in response to the cyclic read delete request.	It is a PLC or DAServer configuration error.	Turn on CYCLIC_TRACE in the Wonderware Logger to obtain additional trace information. If the problem persists, contact Wonderware Technical Support.

Logger Message	Explanation	Probable Cause	Solution
Internal state error: cyclic read response (S7Type 0x%X)	The server receives a duplicate response for the cyclic message.	This is an unknown error.	Turn on CYCLIC_TRACE in the Wonderware Logger to obtain additional trace information. If the problem persists, contact Wonderware Technical Support.
Scan event header key not: 120A: %02X%02X	When parsing the packet returned by the Scan event packet from the PLC, the header key is incorrect.	This is a programming error.	Turn on ALARMS_AND_EVENTS_TRACE in the Wonderware Logger to obtain additional trace information. Repeat the test and/or restart the DAServer/PLC. If the error persists, contact the PLC vendor.
Item doesn't fit in a single message and won't be advised	The server cannot fit this item into a message, therefore it cannot advise this item.	The item byte range is larger than the PLC PDU size.	Split the item into smaller items so that they can be fitted into the available PDU size.
Can't create poke message for item %s, data size is too large	The server cannot fit this item into a poke message, therefore it cannot poke this item.	The item byte range is larger than the PLC PDU size.	Split the item into smaller items so that they can be fitted into the available PDU size.
Can't create poke message for item %s, not able to generate data	The poke data for creating a poke message cannot be generated.	The poke data value cannot be converted into the PLC datatype.	Check the value being poked and create the correct format.
Leaving Slow Poll Mode	This is only a piece of information about the server leaving the Slow Poll mode.	The connectivity to the PLC recovered from a failure. Normal communications is resumed.	If this message shows up consistently, verify the network connectivity to the PLC.

Logger Message	Explanation	Probable Cause	Solution
Entering Slow Poll Mode	This is only a piece of information about the server entering the Slow Poll mode.	The connectivity to the PLC failed. The DAServer tries to reconnect at the Slow Poll interval.	Verify the network connectivity to the PLC. Turn on CONNECTION_TRACE, DASSend, and DASReceive in the Wonderware Logger to obtain additional diagnostic information.
TIMEOUT for pending initiate request	A timeout occurs while waiting for an initiate confirmation.	A communications/ configuration error occured.	Check the communications or configuration. If the problem persists, turn on CONNECTION_TRACE, DASSend, and DASReceive in the Wonderware Logger to obtain additional diagnostic information.
(%d):s7_initiate_req [0]	There is an error in initiating a request (establishing a connection).	A communications/ configuration error occured.	Check the connection and the PLC configuration/ program. If the problem persists, turn on CONNECTION_TRACE, DASSend, and DASReceive in the Wonderware Logger to obtain additional diagnostic information.

Logger Message	Explanation	Probable Cause	Solution
Can't connect	There is an error establishing a connection.	A communications/ configuration error occured.	Check the connection and the PLC configuration/program. If the problem persists, turn on CONNECTION_TRACE, DASSend, and DASReceive in the Wonderware Logger to obtain additional diagnostic information.
Received order id on a deleted non read/write (type: 0x%04X) message: 0x%x	The server receives a response from the PLC for a message that has already been deleted from the server.	Unexpected order ID is received from the PLC. The DAServer discards the message associated with it.	If the problem persists, contact the PLC vendor.
Internal HEXASCII conversion buffer overflow	The server cannot allocate memory for the HEXASCII conversion.	The server may be out of memory.	There could be a memory contention issue on the computer. Check the memory allocations for all processes on the computer.
Generate data failed for item %s because at least one element of the array is not filled	The server cannot poke the array item as the items are not filled correctly.	You poke the array items but some of the element are left unfilled. All elements in the array are rejected for poking.	Fill all elements in the array before poking the array.
TIMEOUT for connection (while %s), m_state=%d	A timeout occurs while waiting for a response message to <command>.	A communications/ configuration error occured.	Verify the network connectivity to the PLC. Turn on CONNECTION_TRACE, DASSend, and DASReceive in the Wonderware Logger to obtain additional diagnostic information.

Logger Message	Explanation	Probable Cause	Solution
(%d):s7_msg_initiate_req (orderid=%d) for %s [(%d) %s]	There is an error in the message initiate request (initiating alarms and events).	A communications/ configuration error occured.	Verify the network connectivity to the PLC. Turn on CONNECTION_TRACE, DASSend, and DASReceive in the Wonderware Logger to obtain additional diagnostic information.
ERROR: order ID %d exists for cyclic ID: %d in cyclic reference map	The server tries to add the order ID for the cyclic service that has already been occupied by some other message.	This is an internal error.	If the problem persists, turn on CYCLIC_TRACE in the Wonderware Logger to obtain additional diagnostic information.
ERROR: Retrieving order ID: %d from cyclic ID: %d in cyclic reference map	The server cannot find the order ID in the order ID map for this message.	This is an internal error.	If the problem persists, turn on CYCLIC_TRACE in the Wonderware Logger to obtain additional diagnostic information.
ERROR: UNKNOWN cyclic ID (%d) in cyclic read indication	The server receives an unknown cyclic message form the PLC.	The PLC reports a cyclic response that is not requested by the DAServer.	Check with the PLC vendor.
Connection aborted	The connection to the PLC is closed.	Either the PLC closed the connection or the server closed the connection.	If this is caused by the normal shutdown or items removal, no actions are required. If not, verify the PLC configuration for the Keep-Alive parameter.

S7 Trace Messages

The SIDirect DAServer provides five types of trace messages as follows:

- Connection Trace
- Poll Trace
- Cyclic Trace
- Block Trace
- Alarms and Events Trace

The following table lists the trace messages produced by the DAServer. For more information about trace messages, see SIDirect DAServer Logger Flags on page 117.

Logger Message	Explanation	Probable Cause	Solution
CONNECTION_TRACE Messages			
(%d):s7_initiate_req [0]	There is an error establishing a connection. The server cannot send the Connect Request to the PLC. The first parameter is the error code of the function call (-1 = Message Blocked, -2 = Message Failed) from the PLC. The parameters inside the square bracket is 0.	The credit is not available to send this request.	If after some time the problem still exists, restart the server.
Can't connect	There is an error establishing a connection. The server cannot send the Connect Request to the PLC.	The credit is not available to send this request.	If after some time the problem still exists, restart the server.
Close connection (con=%s): (cpd=%d,cref=%d)	INFO: The DAServer closes the connection with the PLC. The parameters associated with the PLC connection are listed.	N/A	N/A

Logger Message	Explanation	Probable Cause	Solution
Connection (con=%s) was not successful	The server connection attempts to the PLC fails.	The PLC is in a faulty condition or the connection is broken.	Check the PLC or check the cable connection.
Connection (con=%s) was successful	INFO: The server connects with the PLC successfully.	N/A	N/A
Internal Error: Set state of connection to %d	The connection with the PLC cannot be established.	This is an unknown error.	Check the PLC or cable connection, or restart the server.
Open connection (con=%s)	INFO: The socket connect call to the PLC is successful but the connection negotiation is still not completed	N/A	N/A
s7_abort = OK	INFO: The SIDirect DAServer closes the connection with the PLC successfully.	The connection is closed by either the server or the PLC	N/A
s7_get_initiate_cnf cnf amq called: %d	INFO: PLC agrees on this parameter with the DAServer. (cnf_amq_called)	N/A	N/A
s7_get_initiate_cnf cnf amq calling: %d	INFO: PLC agrees on this parameter with the DAServer. (cnf_amq_calling)	N/A	N/A
s7_get_initiate_cnf cnf pdu size: %d	INFO: PLC agrees on this parameter with the DAServer. (cnf_pdu_size). The PDU size is returned.	N/A	N/A
s7_get_initiate_cnf ind amq called: %d	INFO: This is the DAServer side negotiation parameter with the PLC. (ind_amq_called)	N/A	N/A

Logger Message	Explanation	Probable Cause	Solution
s7_get_initiate_cnf ind amq calling: %d	INFO: This is the DAServer side negotiation parameter with the PLC. (ind_amq_calling).	N/A	N/A
s7_get_initiate_cnf ind pdu size: %d	INFO: This is the DAServer side negotiation parameter with the PLC. (ind_pdu_size) The PDU size of the PLC is requested.	N/A	N/A
s7_initiate_req = OK	The server sends the Connect Request to the PLC successfully.	N/A	N/A
Set state of connection to CLOSED	The SIDirect DAServer closes the connection to the PLC.	The connection is closed by the server or by the PLC.	If the cable is not disconnected, check the PLC configuration and see if you have specified the Keep-Alive parameter. The Keep-Alive parameter causes the closing of connection if there are no activities for some specified amount of time.
Set state of connection to ERROR	The connection with the PLC cannot be established.	The server cannot connect to the PLC at all.	Check the PLC or cable connection, or restart the server.
Set state of connection to OPEN	INFO: The server opens a connection with the PLC.	N/A	N/A
Set state of connection to PENDING	INFO: The server is awaiting the response from the PLC for the Connection Request.	N/A	N/A

Logger Message	Explanation	Probable Cause	Solution
Set state of connection to SLOW POLL MODE	The server is going into the Slow Poll mode.	The connection with the PLC is bad.	Check the PLC or check the cable connection.
POLL_TRACE Messages			
%s dumping s7 objects (containing %d items)	INFO: Dumping the information about the S7Info in the logger.	N/A	N/A
(1):s7_multiple_read _req (orderid=%d) [0]	INFO: The multiple read request message cannot be sent to the PLC because the server runs out of credit.	N/A	N/A
(2):s7_multiple_read _req (orderid=%d) [0]	The multiple read request message cannot be sent to the PLC because of unknown reasons.	N/A	If the problem persists, restart the server.
add pItem (%s) to S7Info (pS7Info=0x%08X, addr: %d range: %d)	INFO: The server adds item to the S7Info while building the Poll message.	N/A	N/A
Build message (%s msg=0x%08X) for topic %s	INFO: The server builds a POLL message for the topic.	N/A	N/A
Delete message (%s msg=0x%08X) for topic %s	INFO: Destructor for the POLL message gets called.	N/A	N/A
dumping read values	INFO: Dumping the read values received from the PLC in the logger.	N/A	N/A
Got orderid:%d for message (%s msg=0x%08X,con=%s) [number of orderids=%d]	INFO: The server assigns the order ID for the POLL message.	N/A	N/A
Release orderid:%d for message (%s msg=0x%08X,con=%s) [number of orderids=%d]	INFO: Release the order ID from the message.	N/A	N/A

Logger Message	Explanation	Probable Cause	Solution
remove pItem (%s) from S7Info (pInfo=0x%08X)	INFO: The server removes the item from the S7Info.	N/A	N/A
s7_multiple_read_req = OK (orderid=%d)	INFO: The multiple read request message is sent to the PLC successfully.	N/A	N/A
S7Info=%s:result=%d var_length=%d value (in HEXASCII):	INFO: The server logs the information about the read value received from the PLC in the logger.	N/A	N/A
POKE_TRACE Messages			
%s dumping s7 objects (containing %d items)	INFO: Dumping the information about the S7Infos in the logger.	N/A	N/A
%s: result=%d var_length=%d value (in HEXASCII)	INFO: Logging the Result received from the PLC for the write request.	N/A	N/A
(1):s7_multiple_write_req (orderid=%d) [(%d) %s]	INFO: The multiple write request message cannot be sent to the PLC because running out of credit.	N/A	N/A
(2):s7_multiple_write_req (orderid=%d) [(%d) %s]	The multiple write request message cannot be sent to the PLC because of unknown reasons.	N/A	If problem persists, restart the Server or check the PLC.
add pItem (%s) to S7Info (pS7Info=0x%08X, addr: %d range: %d)	INFO: Server adding item to the S7Info while building the Poke Message.	N/A	N/A
Build message (%s msg=0x%08X) for topic %s	INFO: Server building a Poke message for the topic.	N/A	N/A
Delete message (%s msg=0x%08X) for topic %s	INFO: Destructor for the Poke message gets called.	N/A	N/A

Logger Message	Explanation	Probable Cause	Solution
Got orderid:%d for message (%s msg=0x%08X,con=%s) [number of orderids=%d]	INFO: Server assigns the order id for the Poke Message.	N/A	N/A
Release orderid:%d for message (%s msg=0x%08X,con=%s) [number of orderids=%d]	INFO: Release the order ID from the poke message.	N/A	N/A
remove pItem (%s) from S7Info (pInfo=0x%08X)	INFO: Server removing the item from the S7Info.	N/A	N/A
s7_multiple_write_req = OK (orderid=%d)	INFO: The multiple write request message sent to the PLC successfully.	N/A	N/A
S7PokeMessage::AddPokeItem (Item=%s)	INFO: Server adding item to the Poke Message.	N/A	N/A
TREAL item %s high clamped to 9990s	INFO: The value of TREAL item is clamped at high value.	N/A	N/A
TREAL item %s low clamped to 0ms	INFO: The value of TREAL item is clamped at low value.	N/A	N/A
CYCLIC_TRACE Messages			
%s dumping s7 objects (containing %d items)	INFO: Dumping the information about the S7Infos in the logger.	N/A	N/A
(-1):s7_cycl_read (orderid=%d) [(%d) %s]	The server could not send the cyclic read request to the PLC. Temporary error.	N/A	N/A
(1):s7_cycl_read_delete_req (orderid=%d): [0]	The server could not send the cyclic read delete request to the PLC because of running out of credit.	N/A	N/A

Logger Message	Explanation	Probable Cause	Solution
(1):s7_cycl_read_start_req (orderid=%d): [0]	The server could not send the cyclic read start request to the PLC because of running out of credit.	N/A	N/A
(2):s7_cycl_read_delete_req (orderid=%d): [0]	The server could not send the cyclic read delete request to the PLC because of Unknown reason.	N/A	N/A
add pItem (%s) to S7Info (pS7Info=0x%08X, addr: %d range: %d)	INFO: Server adding item to the S7Info while building the cyclic Message.	N/A	N/A
Build message (%s msg=0x%08X) for topic %s	INFO: Server building a cyclic message for the topic.	N/A	N/A
Delete message (%s msg=0x%08X) for topic %s	INFO: Destructor for the cyclic message gets called.	N/A	N/A
Got orderid:%d for message (%s msg=0x%08X,con=%s) [number of orderids=%d]	INFO: Server assigns the order id for the cyclic Message.	N/A	N/A
Release orderid:%d for message (%s msg=0x%08X,con=%s) [number of orderids=%d]	INFO: Release the order ID from the cyclic message.	N/A	N/A
remove pItem (%s) from S7Info (pInfo=0x%08X)	INFO: Server removing the item from the S7Info.	N/A	N/A
s7_cycl_read = OK (orderid=%d)	INFO: The server successfully sends the cyclic read request to the PLC.	N/A	N/A
s7_cycl_read_delete_req = OK (orderid=%d)	INFO: The server successfully sends the cyclic read delete request to the PLC.	N/A	N/A

Logger Message	Explanation	Probable Cause	Solution
s7_cycl_read_start_req = OK (orderid=%d)	INFO: The server successfully sends the cyclic read start request to the PLC.	N/A	N/A
s7_get_cycl_read_init_cnf (cpd=%d,m_cref=%d, orderid=%d) Build Poll Messages: [(%d) %s]	INFO: The server gets the confirmation from the PLC for the Cyclic read request.	N/A	N/A
S7Info=%s:result=%d var_length=%d value (in HEXASCII)	INFO: Logging the value received from the PLC for the cyclic message	N/A	N/A
BLOCK_TRACE Messages			
(0): s7_brcv_init (r_id=%d) [0]	INFO: The block receive initiation request is successful.	N/A	N/A
add item (%s) to S7BlockMessage (%s msg=0x%08X)	INFO: Adding items to the block message.	N/A	N/A
Build message (%s msg=0x%08X) for topic %s	INFO: Server building a Block message for the topic.	N/A	N/A
Delete message (%s msg=0x%08X) for topic %s	INFO: Destructor for the block message gets called.	N/A	N/A
Release blockid:%d for message (%s msg=0x%08X,con=%s) [number of blockids=%d]	INFO: Release block id for the block message.	N/A	N/A
remove item (%s) from S7BlockMessage (%s msg=0x%08X)	INFO: Remove item from the block message.	N/A	N/A
s7_brcv_init = OK (r_id=%d)	INFO: The block receive initiation request is successful.	N/A	N/A
s7_brcv_stop = OK (r_id=%d)	INFO: The block receive stop request is successful.	N/A	N/A

Logger Message	Explanation	Probable Cause	Solution
ALARMS_AND_EVENTS_TRACE Messages			
%s: ack_state=%d	INFO: Update the alarm or event with the acknowledgement state information.	N/A	N/A
%s: event_state=%d	INFO: Update the alarm or event with the event state information.	N/A	N/A
%s: no_add_value=%d	INFO: Update the alarm or event with the number of additional value information.	N/A	N/A
%s: state=%d	INFO: Update the alarm or event with the state information.	N/A	N/A
%s: time_stamp=%s	INFO: Update the alarm or event with the time stamp information.	N/A	N/A
(%d):s7_msg_abort_req (orderid=%d) for %s [0]	The alarm abort request to the PLC failed.	N/A	N/A
add item (%s) to S7Event (msg=0x%08X)	INFO: Add item to the event message.	N/A	N/A
Build alarm object (0x%08X) for connection %s	INFO: Server builds the alarm object that are going to receive the alarms and events.	N/A	N/A
Build scan object (0x%08X) for connection %s	INFO: Server builds the scan object.	N/A	N/A
Delete alarm object (0x%08X) for connection %s	INFO: The destructor of the alarm object gets called.	N/A	N/A
Release eventid:%d (con=%s) [number of eventids=%d]	INFO: Release order id from the event message.	N/A	N/A

Logger Message	Explanation	Probable Cause	Solution
remove item (%s) from S7Event (0x%08X)	INFO: Remove item from the event message.	N/A	N/A
s7_msg_abort_req = OK (orderid=%d) for %s	INFO: The server sends the alarm abort request to the PLC successfully.	N/A	N/A
s7_msg_initiate_req = OK (orderid=%d) for %s	INFO: The server sends the alarm registration request to the PLC successfully.	N/A	N/A

DAServer Error Codes

The following table lists the Wonderware DAServer error codes and the error messages that appear with the codes, and their descriptions.

Code	Error Message	Description
C004D000L	Invalid item name	The requested item name has bad syntax.
C004D001L	Item name not exist	The requested item name has good syntax, but it does not exist.
C004D002L	Device not connect	The device is not connected, so data cannot be acquired.
C004D100L	Device off scan	The device is communicating, but it cannot accept queries for data items.
C004D101L	Timeout	A message transaction with the device timed out.

DAServer Protocol Warnings

The following table lists protocol warnings generated by the DAServer. The log flag for these messages is DASProtWarn.

Logger Message	Explanation	Probable Cause	Solution
DASProtWarn Messages			
Send: exit (MSG_OK): Attempt to send POLL message while pending (%s msg=0x%08X) [msg_state=%d,con_state=%d]	The server tries to send the Poll message while it waits for the response from the PLC for the same message.	The response from the PLC is slow.	This is a flow control issue. The server is too fast to send the message but the PLC is slow in responding to those messages. Try to reduce the load from the PLC by disconnecting other clients from the PLC or reducing the scan rate of the message. If the problem exists call the PLC vendor.
Update item (%s, quality=0x%04X) on %s	The server updates the item with Bad quality. This message shows up only when the item quality is Bad.	N/A	Check the OPC quality for the appropriate error message.
S7Info (%s pS7Info=0x%08X/p Msg=0x%08X) returned error: %s	Logging the error code returned by the PLC for the read request.	There is an item access error in the PLC.	Check the PLC configuration and see whether the memory area you try to access exists in the PLC with proper access right.
Could not generate data for item %s	The server cannot read the poke value.	This is an internal error.	Turn on POKE_TRACE in the Wonderware Logger to obtain additional trace information. Report the error to Wonderware Technical Support.

Logger Message	Explanation	Probable Cause	Solution
Send: exit (MSG_OK): Attempt to send POKE message while pending (%s msg=0x%08X) [msg_state=%d,con_state=%d]	The server tries to send the poke message while it waits for the response from the PLC for the same message.	The response from the PLC is slow.	This is a flow control issue. The server is too fast to send the message but the PLC is slow in responding to those messages. Try to reduce the load from the PLC by disconnecting other clients from the PLC or reducing the scan rate of the message. If the problem exists call the PLC vendor.
S7 Topic's <%s> property <%s> was changed to <%s>	The server is not hot-configurable for the given property.	The server does not use the changed value.	Re-start the server to see this change in effect.
Invalid value, clamp at high limit for poking item: %s on %s	Poke data is clamped into a valid range before it is sent to the PLC.	Poke value exceeds the S7 data type range.	See the DAServer user's guide for the correct range of values.
Clamping S5T poke data for %s on %s (client poke %ums clamped to 9990000)	S5T poke value is clamped to 9990000 before it is sent to the PLC.	Poke value exceeds the valid range.	See the DAServer user's guide for the correct range of values.
Loosing precision on converting S5T poke data for %s on %s (client poke %ums converted to 0ms)	Non-zero S5T poke value is converted to 0.	Poke value is below the S7 S5T type resolution.	See the DAServer user's guide for the correct range of values.
Loosing precision on converting S5T poke data for %s on %s (client poke %ums converted to %ums)	Precision is lost on value.	The resolution of the S7 data type does not match the poked value.	See the DAServer user's guide for the correct range of values.
Invalid poke value, clamp at low limit for item: %s on %s	Poke data is clamped into a valid range before it is sent to the PLC.	Poke value exceeds the S7 data type range.	See the DAServer user's guide for the correct range of values.

Logger Message	Explanation	Probable Cause	Solution
Invalid poke value, cannot convert value for item: <item name> on <device group>	Poke data is set to constant.	Poke value exceeds the S7 data type range.	See the DAServer user's guide for the correct range of values.
ERROR: Invalid value, clamp at high limit for poking item: %s on %s	Poke data is clamped into a valid range.	Poke value exceeds the S7 data type range.	See the DAServer user's guide for the correct range of values.
ERROR: Invalid value, clamp at low limit for poking item: %s on %s	Poke data is clamped into a valid range.	Poke value exceeds the S7 data type range.	See the DAServer user's guide for the correct range of values.
ERROR: Invalid value, cannot convert for poking item: %s on %s	Poke data is set to constant.	Poke value exceeds the S7 data type range.	See the DAServer user's guide for the correct range of values.
Write complete fails - item: %s on %s	The server cannot write the value of the item to the PLC.	Connection to the PLC is bad or Item access is denied by the PLC.	Check the PLC connection or configuration.
S7Cp's <%s> property <%s> was changed to <%s>	The server is not hot-configurable for the given property.	The server does not use the changed value.	Restart the server to see this change in effect.
(%d):s7_msg_initiate_req (orderid=%d) for %s [(0)]	There is an error in the message initiate request (initiating alarms and events).	It is a communications error or a PLC configuration error.	Check the connection and the PLC configuration/program.
S7CP_200's <%s> property <%s> was changed to <%s>	The server is not hot-configurable for the given property.	The server does not use the changed value.	Restart the server to see this change in effect.

Data Conversion

The following table describes how the SIDirect DAServer handles values that cannot be converted or do not meet the limit specifications.

Conversion	Description
NONSPECIFIC	If a value cannot be converted, the quality of the item goes to NONSPECIFIC.
Uncertain-HIGHLIMITED	If a value is greater than the upper limit, the quality of the item goes to uncertain-HIGHLIMITED.
Uncertain-LOWLIMITED	If a value is minor than the lower limit, the quality of the item goes to uncertain-LOWLIMITED.

Quality Settings

The SIDirect DAServer uses the general OPC-defined quality settings. An item can have six basic data quality states.

Quality Code	Quality State	Description
00C0	Data quality good	Data communications is good and data is good.
		The register is read or written to without any problems converting the data.
0055	Clamp low	Data communications is good but the data is uncertain.
		The data is clamped at a low limit.
		The register is correctly read or written to, but it is necessary to clamp its value to a limit.
		The value is smaller than the minimum allowed.

Quality Code	Quality State	Description
0056	Clamp high	Data communications is good but the data is uncertain.
		The data is clamped at a high limit.
		The register is correctly read or written to, but it is necessary to clamp its value to a limit.
		The value is larger than the maximum allowed.
		A string is truncated.
		For example, a floating point value is clamped to FLT_MAX.
0040	Quality uncertain/No convert	Data communications is good but the data is uncertain.
		The data cannot be converted.
		The server may return either a constant in place of the data or return quality information alone.
		The data is useable. However, it is not known whether the value is too large or too small.
		Incorrect data type.
		Floating point is not a number.
		For example, 0x000a in a PLC BCD register.

Quality Code	Quality State	Description
0004	Bad configure/No access	This is a configuration error.
		Data communications is good but the data cannot be sent and/or received. The data is bad and cannot be used.
		Item cannot be accessed.
		The item does not exist or is not available.
		The server can communicate with the PLC but cannot access the register.
		The server determined the point is not valid.
		The PLC responds that the register does not exist, cannot be read, or cannot be written to.
		The server cannot access a fenced, write-protected, or read-only item.
		The PLC is in a mode that does not permit access to this item.
		The number of data bytes is incorrect but the message is otherwise good.
		The command or op code is invalid but the message is otherwise good.
		The PLC is busy. The server has given up retrying.

Quality Code	Quality State	Description
0018	No communications	Data communications is down.
		Cannot access the PLC due to a communications error.
		Data is bad and cannot be used.
		The device group is in a slow poll or equivalent mode.
		The PLC does not exist and/or is not responding.
		There is no link validating the message.
		There is a lack of resources in the server. A TSR or driver cannot allocate memory.
		There is a lack of resources in the communications link.
		The communications link is off-line.
		All communications channels are in use.
		The network cannot route the message to the PLC.

Chapter 5

Reference

This section describes the architecture of the DAServer, its collection of components, and the environments in which the components work.

DAServer Architecture

The SIDirect DAServer is supported on Microsoft Windows 2003 Standard and Enterprise Editions, Windows XP Pro, Windows 2000 Professional, Windows 2000 Server, and Windows 2000 Advanced Server. The NetDDE protocol is not supported by DAServers.

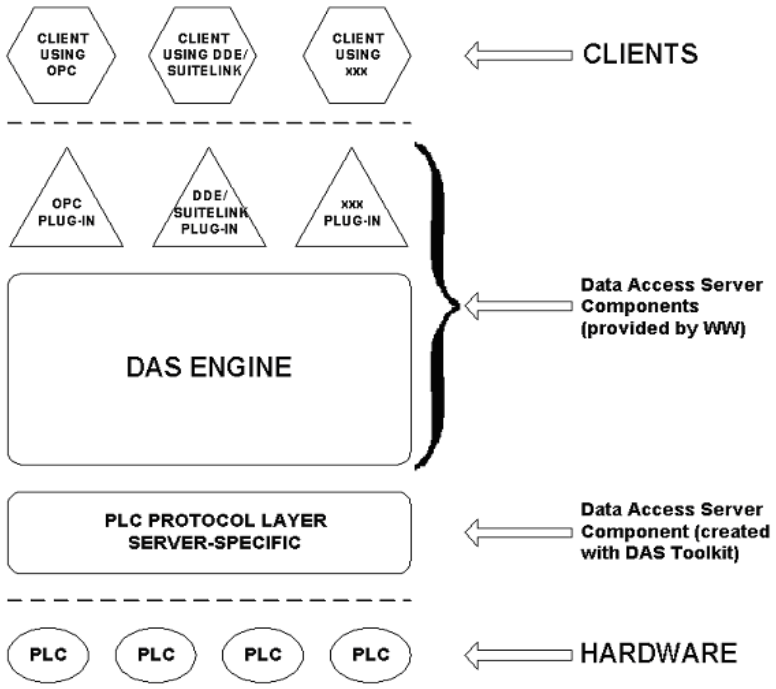
This DAServer is a collection of components that work together to provide communications access to the hardware field devices. These components include:

- **DAServer Manager**
  The Microsoft Management Console (MMC) snap-in environment, supplied with the DAServer, that provides the necessary user-interface for diagnostics, configuration, and activation.
- **Client Plug-ins**
  The components that are added to a DAServer to enable communications with clients.
  Examples are OPC, DDE/Suitelink, and so on.
- **DAS Engine**
  The library that contains all the common logic to drive data access.
- **Device Protocol**
  The custom code provided by the DAServer to define the communications with a particular device.

DAServers

A DAServer consists of the following three physical parts:

- **Plug-in Component(s)**
  This component 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 consists of a set of .exe and/or .dll modules. Wonderware provides the plug-ins and the DAS Engine. Using the DAS Toolkit, you create 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 in an array are not supported in the DDE/SL plug-in. Arrays are converted to HEXASCII strings, which provide legacy behavior for DAServers that support this.

DAS Engine

The DAS Engine is a middleware component that exposes two sets of interfaces, one for communicating with the plug-ins and the other 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. A new DAS Engine such as feature enhancements or bug fixes do not require relinking to the other components nor re-QA of those other components. When deployed to the system, the new DAS Engine attaches to all existing DAServer components.
- Newly deployed plug-ins such as feature enhancements or bug fixes do not require relinking nor re-QA of associated components. Even new plug-ins (for example, OPC Alarm & Events) do not require any development changes to the other components, and require no relinking in a customer-installed base. In fact, you can implement new functionality in a plug-in to enhance the DAServer without involving the code of the other components.
- DAServers can be configured in one stand-alone configuration utility (DAServer Manager) capable of showing specific configuration pages for all DAServers. This utility allows the browsing and editing of DAServers on different nodes.
- The DAServer Manager diagnostics tool shows 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 such as, XML Editor can read this format.

Appendix A

Supported DAServer Hardware and Firmware

The following table lists the hardware and firmware supported by the SIDirect DAServer Version 1.5.

Device	Description	Hardware and Firmware
S7-200	Controller	CP243-1 TCP 243-1EX00-0XE0 Hardware Version: 1.0 Firmware Version: 02.00
S7-300	Controller	CP343-1 343-1EX11-0XE0 Hardware Version: 2 Firmware Version: 2.0.0
		CP343-1 343-1EX11-0XE0 Hardware Version: 2 Firmware Version: 2.1.5
		CP343-5 343-5FA01-0XE0 Hardware Version: 4 Firmware Version: 4.1.3
	Built-in Ethernet port	CPU317-2DP/PN Hardware Version: 1.0 Firmware Version: 2.3.2

Device	Description	Hardware and Firmware
S7-400	Controller	CP443-1 TCP 443-1EX00-0XE0 Hardware Version: 5 Firmware Version: 3.00
		CP443-1 TCP 443-1EX00-0XE0 Hardware Version: 6 Firmware Version: 3.00
		CP443-1 443-1EX02-0XE0 Hardware Version: 2 Firmware Version: 5.1.0
S7-400	Controller	CP443-1 443-1EX11-0XE0 Hardware Version: 2 Firmware Version: 1.1.0
		CP443-1 443-1EX11-0XE0 Hardware Version: 4 Firmware Version: 2.1.0
		CP443-1 443-1EX11-0XE0 Hardware Version: 4 Firmware Version: 2.2.17
		CP443-1 443-1EX11-0XE0 Hardware Version: 4 Firmware Version: 2.2.5

Index

S

T

U

V

W

X