

# Wonderware Operations Integration - Supervisory SIdirect Server (G-1.1 Series)





# Contents

Chapter 1 Introduction to the SIDirect OI Server.....	9
About the SIDirect OI Server .....	9
SIDirect Legacy Object: Non-Symbolic Namespace .....	9
SIDirect Symbolic Object: Symbolic Namespace .....	10
Supported Client Protocols .....	10
OPC .....	10
SuiteLink .....	10
DDE/FastDDE .....	10
DDE .....	11
FastDDE .....	11
Supported Device Protocols .....	11
Chapter 2 SIDirect OI Server Configuration .....	13
SIDirect OI Server Installation.....	13
SIDirect Preconfigured Hierarchy .....	14
SIDirect Legacy Object Pre-Configuration.....	16
Configuring the SIDirect Legacy Object .....	17
SIDirect Legacy Object Parameters .....	19
SIDirect Legacy Device Groups and Device Items .....	20
Exporting and Importing SIDirect OI Server Item Data .....	23
Configuring the SIDirect Symbolic Object .....	24
SIDirect Symbolic Object Parameters .....	25
SIDirect Symbolic Device Groups and Device Items .....	26
Device Redundancy .....	28
Run-time Behavior of Redundant Devices .....	29
Configuring Device Redundancy .....	29
SIDirect Scan-Based Message Handling.....	32
Chapter 3 SIDirect OI Server Reference.....	33
SIDirect Legacy Object Reference (Absolute Addressing) .....	33
SIDirect Absolute Naming Convention .....	33
S7-1500 Item Syntax .....	34
S7-1500 Item Naming.....	34
S7-1500 Data Blocks and Instance Blocks.....	34
Flag Bytes .....	37
Input Bytes .....	41
Output Bytes .....	45
Peripheral Input Bytes .....	50
Peripheral Output Bytes.....	53
Block Items .....	58
Read-Only Block Items.....	58
Write-Only Block Items .....	61

Counters .....	63
High-Speed Counters .....	64
LREAL Data Type and Syntax.....	64
DTL Data Type and Syntax .....	66
DTL Clamping.....	69
Using DTL without a Suffix .....	70
Using DTL with a TREAL Suffix .....	70
Conversions and Suffixes of Items (Absolute Addressing) .....	70
Endian Conversion .....	70
Suffix BCD .....	70
Suffix DT .....	70
Suffix KT .....	71
Suffix S5T .....	71
Suffix TR.....	71
Suffix D .....	72
Suffix T.....	72
Suffix TOD .....	72
Generic OPC Syntax .....	72
VT_Array Syntax in Application Server .....	73
SIDirect Symbolic Object Reference (Symbolic Addressing) .....	73
SIDirect Symbolic Naming Convention.....	73
Basic Data Types .....	74
Complex Data Types .....	74
User Defined Data Types .....	74
System Data Types .....	74
General SIDirect Item Syntax Constraints.....	75
Data Types in the TIA Portal.....	75
General Data Types .....	75
Counter/Timer Data Types .....	78
Special Data Types .....	82
Conversions and Suffixes of Items (Symbolic Addressing).....	84
Suffix D (Symbolic) .....	84
Suffix LDT (Symbolic).....	85
Suffix LT (Symbolic) .....	85
Suffix S5T (Symbolic) .....	85
Suffix T (Symbolic) .....	85
Suffix TOD (Symbolic).....	86
SIDirect Driver Diagnostic Info Items.....	86
Data Conversion .....	90
Quality Settings .....	91
Item Validation .....	92
Tested Hardware and Firmware Versions .....	93
<b>Chapter 4 Troubleshooting the SIDirect OI Server.....</b>	<b>95</b>
Troubleshooting Tools .....	95
Finding the SIDirect OI Server Version Number .....	95
Debugging Communications Between the SIDirect OI Server and the PLC .....	95
Client Groups .....	96
Structure .....	98
Transactions .....	99
Statistics.....	99

---

Message View .....	99
Devices Groups .....	101
Diagnostics and Error Tracing .....	102
Diagnostics Facility .....	102
Communications Processor Diagnostics .....	102
S7 Communications Processor Diagnostics .....	102
Items Diagnostics .....	102
Messages Diagnostics .....	103
Device Group Diagnostics .....	103
OI Server (DASTrace) Diagnostic Messages .....	103
Error Tracing with the Wonderware Logger .....	104
SIdirect OI Server Logger Flags .....	104
Error Messages, Trace Messages, Error Codes, and Warnings .....	105
OI Server Error Messages .....	106
S7 Trace Messages .....	113
OI Server Error Codes .....	122
OI Server Protocol Warnings .....	123
Index .....	127

© 2016 Schneider Electric Software, LLC. All rights reserved.

No part of this documentation shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Schneider Electric Software, LLC. No liability is assumed with respect to the use of the information contained herein.

Although precaution has been taken in the preparation of this documentation, Schneider Electric Software, LLC assumes no responsibility for errors or omissions. The information in this documentation is subject to change without notice and does not represent a commitment on the part of Schneider Electric Software, LLC. The software described in this documentation is furnished under a license agreement. This software may be used or copied only in accordance with the terms of such license agreement.

ArchestrA, Avantis, DYNsIM, eDNA, EYESIM, Foxboro, Foxboro Evo, I/A Series, InBatch, InduSoft, InStep, IntelaTrac, InTouch, PIPEPHASE, PRiSM, PRO/II, PROVISION, ROMeo, Schneider Electric, SIM4ME, SimCentral, SimSci, Skelta, SmartGlance, Spiral Software, VISUAL FLARE, WindowMaker, WindowViewer, and Wonderware are trademarks of Schneider Electric SE, its subsidiaries, and affiliated companies. An extensive listing of Schneider Electric Software, LLC trademarks can be found at: <http://software.schneider-electric.com/legal/trademarks/>. All other brands may be trademarks of their respective owners.

Schneider Electric Software, LLC  
26561 Rancho Parkway South  
Lake Forest, CA 92630 U.S.A.  
(949) 727-3200

<http://software.schneider-electric.com/>

Publication date: 7/11/2016

# Contact Us

## Contact Schneider Electric Software Technical Support

Contact our Software Technical Support Group if you have a question or issue with Avantis, SimSci, Skelta, or Wonderware software products.

- Priority email for Customer First Members: [custfirstsupport@wonderware.com](mailto:custfirstsupport@wonderware.com)
- Customers without a support agreement: [wwsupport@wonderware.com](mailto:wwsupport@wonderware.com)

Americas: Lake Forest, California, U.S.

- Tel: + 1 800 966 3371 / + 1 949 639 8500

Asia Pacific: Shanghai, China

- Tel: + 800 4500 3457 (All Asia) / + 400 820 1521 (China) / + 86 400 8201521
- Email: [gcssupport.apac@schneider-electric.com](mailto:gcssupport.apac@schneider-electric.com)

Europe and Middle East, Africa (EMEA): Bratislava, Slovakia

- Tel: + 421 2 32 200 342 / See our website at <http://software.schneider-electric.com/> for a list of toll-free numbers.

## Contact Schneider Electric Software Learning Services

Contact Software Learning Services for assistance regarding classes, schedule, offerings, frequently asked questions, tuition, policies, and more.

- Phone: + 1 866 998 7246 in the U.S. and Canada
- + 1 949 639 8508 direct
- Fax: + 1 949 639 1847
- Email: [software.training@schneider-electric.com](mailto:software.training@schneider-electric.com)





# CHAPTER 1

## Introduction to the SIDirect OI Server

### In This Chapter

About the SIDirect OI Server.....	9
SIDirect Legacy Object: Non-Symbolic Namespace.....	9
SIDirect Symbolic Object: Symbolic Namespace .....	10
Supported Client Protocols .....	10
Supported Device Protocols.....	11

### About the SIDirect OI Server

The SIDirect Operations Integration Server (referred to as the OI Server throughout the remainder of this user's guide) is a Microsoft® Windows® application program that acts as a communications protocol server to provide access to data within the Siemens SIMATIC S7-1500 PLC series.

The OI Server provides access to a Siemens SIMATIC S7-1500 PLC through an off-the-shelf standard Ethernet network interface card in the computer. Multiple network interface cards can be supported by the OI Server.

The SIDirect OI Server connects via Ethernet to the S7-1500 series PLC. It does **not** require any of the following items:

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

The SIDirect OI Server contains two objects that can be configured to interface with the Siemens S7-1500 PLC:

- *Legacy object* (see "*SIDirect Legacy Object: Non-Symbolic Namespace*" on page 9) – supports SIMATIC tag names.
- *Symbolic object* (see "*SIDirect Symbolic Object: Symbolic Namespace*" on page 10) – supports symbolic tag names.

While the SIDirect OI Server can support other PLCs in Siemens S7 family of PLCs, it is specifically designed to support the S7-1500 . For the complete list of supported hardware and firmware, see "Tested OI Server Hardware and Firmware".

### SIDirect Legacy Object: Non-Symbolic Namespace

The SIDirect Legacy object provides an interface to support traditional SIMATIC tag names (for example, "DB1, INT0"). Use the Legacy object if you are using non-symbolic names (absolute addressing) to communicate with the PLC. To use absolute addressing, check the following settings in the S7\_1500 PLC:

- PUT/GET communication must be enabled.

- Optimized block access must be disabled (non-optimized mode).

See *SIDIrect Legacy Object Pre-Configuration* on page 16 for additional information.

## SIDIrect Symbolic Object: Symbolic Namespace

The SIDIrect Symbolic object provides an interface to support symbolic tag names (for example, "Tank4.Lvl"). Use the Symbolic object if you are using symbolic names to communicate with the PLC. Symbolic addressing works whether or not optimized block access has been enabled in the S7-1500 PLC.

## Supported Client Protocols

The OI Server uses the following different communications protocols to communicate with clients:

- OPC
- SuiteLink
- DDE/FastDDE

### 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 operating systems.

SuiteLink is not a replacement for DDE or FastDDE. 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.

### DDE/FastDDE

DDE/FastDDE communication protocols allow communication between a client and a server. DDE protocol is developed by Microsoft whereas FastDDE protocol is proprietary to Wonderware.

## DDE

DDE is a communications protocol that allows 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 the InTouch HMI or Microsoft Excel, can simultaneously be both a client and a server.

---

**Important:** Local DDE is supported only when the OI Server is configured as "Not a Service" and activated from its executable file or started from the InTouch HMI. Local DDE is not supported when the OI Server is activated from the System Management Console (SMC).

---

## 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.

## Supported Device Protocols

The SIDirect OI Server uses only the TCP bus communications protocols over the Ethernet to communicate with the Siemens S7-1500 family of controllers. The SIDirect OI Server does not support MPI, Profibus, and other non-Ethernet protocols.



# CHAPTER 2

## SIDirect OI Server Configuration

### In This Chapter

SIDirect OI Server Installation .....	13
SIDirect Preconfigured Hierarchy .....	14
Device Redundancy .....	28
SIDirect Scan-Based Message Handling .....	32

### SIDirect OI Server Installation

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

Reading the documentation for both the MMC and the OI Server Manager is critical to understanding this user's guide. To read the documentation about the MMC and OI Server Manager, click the Help topics on the MMC **Help** menu. Both the MMC and OI Server Manager Help are displayed.

---

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

---

#### To install the Wonderware SIDirect OI Server:

1. Run the command prompt as administrator.
2. Navigate to the SIDirect OI Server CD Setup directory.
3. Type **setup.bat** and press **Enter**.
4. 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 **Operations Integration Server Manager**.

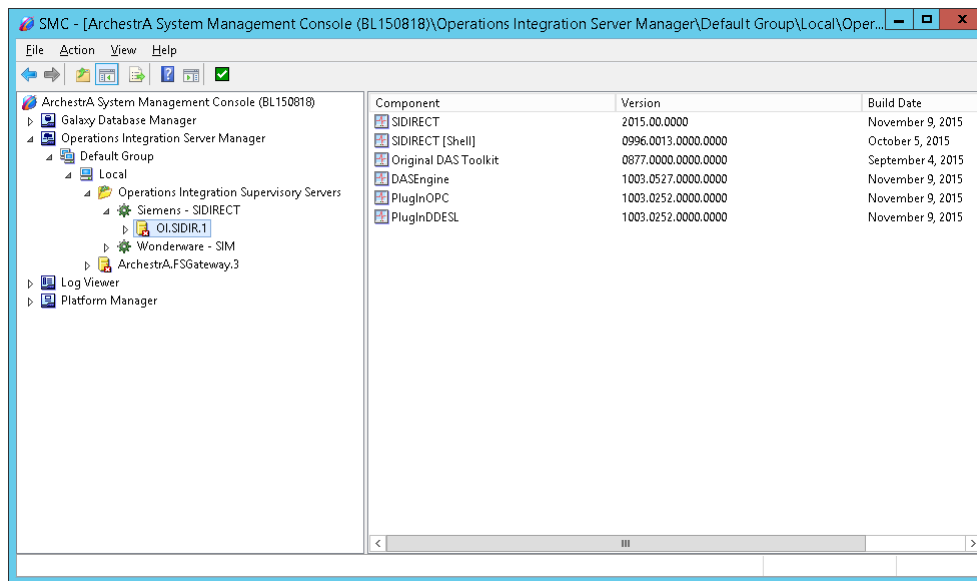
---

**Note:** See the Operations Integration Server Manager documentation for general information about working in the System Management Console (SMC) snap-in environment.

---

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

6. Click on the **Operations Integration Supervisory Servers** folder. Installed OI Server drivers are listed under the folder. The driver name is **Siemens - SIDirect**. The SIDirect OI Server is listed under the driver name: **OI.SIDIR.1**.



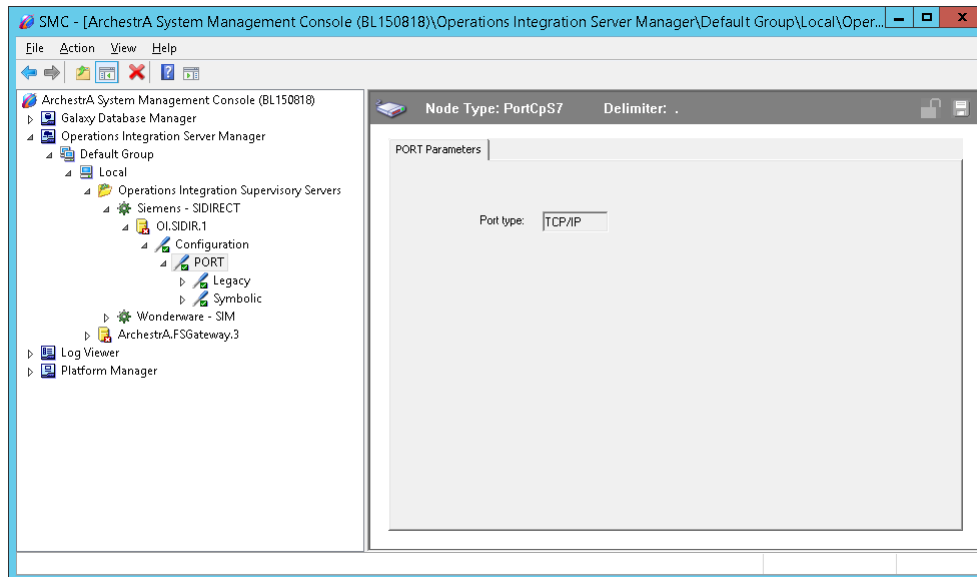
## SIDirect Preconfigured Hierarchy

The server-specific configuration portion of the SIDirect OI Server hierarchy tree under the OI Server Manager starts at the Port 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

When you install the Wonderware SIDirect OI Server, it is installed with a preconfigured device hierarchy in the **ArchestrA System Management Console**. The SIDirect OI Server uses a two-tier hierarchy for modeling the S7-1500 objects. This hierarchy consists of a Port object and two CPU objects:

- **PORT:** This is the CpS7 communications port object that communicates with a CpS7 node.
- **Legacy Object:** The Legacy object is subordinate to the PORT object. Use the legacy object to support absolute addressing (for example, "DB1, INT0"). Optimized Block Access must be disabled (non-optimized mode) in the S7-1500 PLC in order to use absolute addressing.
- **Symbolic Object:** The Symbolic object is subordinate to the PORT object, and works with symbolic addressing to permit the use of enhanced tag names (for example, "Tank4.Lvl"). Symbolic addressing works whether or not optimized block access has been enabled in the S7-1500 PLC.

To view the preconfigured hierarchy, click **Operations Integration Supervisory Servers -> Siemens - SIDirect -> OI.SIDIR.1 -> Configuration -> Port**



1. Before starting the OI Server, modify the device hierarchy as needed to establish communications to each of the controllers.
  - For step-by-step procedures on how to configure the Legacy Object, see *Configuring the SIDirect Legacy Object* on page 17.
  - For step-by-step procedures on how to configure the Symbolic Object, see *Configuring the SIDirect Symbolic Object* on page 24.

---

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

---

2. Configure the respective devices for communications.
3. Finally, create the Device Groups for each controller. Navigate to the object in the **OI Server Manager** tree view and click on the **Device Groups** tab. For step-by-step procedures on configuring Device Groups, see:
  - *Editing SIDirect Legacy Object Device Group Parameters* on page 20
  - *Editing SIDirect Symbolic Object Device Group Parameters* on page 26

---

**Note:** If a configuration view is in an open state and you open the same server the second time, the OI Server 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 OI Server is ready to use after you activate it.

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

To run the SIDirect OI Server as a service, right-click on the **OI.SIDIR.1** OI Server name and select **Configure As Service**. You can configure it as an auto service or manual service.

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

## SIDirect Legacy Object Pre-Configuration

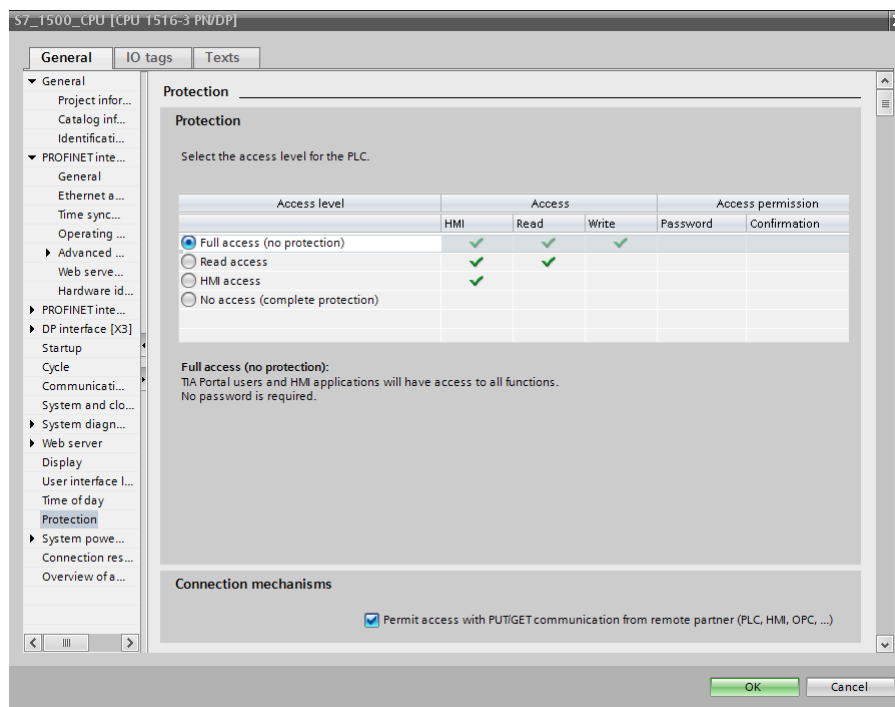
The SIDirect Legacy object requires some pre-configuration through the Siemens TIA portal to allow access to data inside the S7-1500 PLC. This pre-configuration consists of:

- Enabling PUT/GET communication
- Disabling optimized block access

Pre-configuration is not required for the SIDirect Symbolic object.

### To enable PUT/GET communication with the S7-1500 PLC:

1. Open the Siemens TIA portal.
2. In the TIA portal, locate the S7-1500 CPU in the left tree.
3. Right-click the S7-1500 CPU and then click **Properties**. The properties dialog box appears.



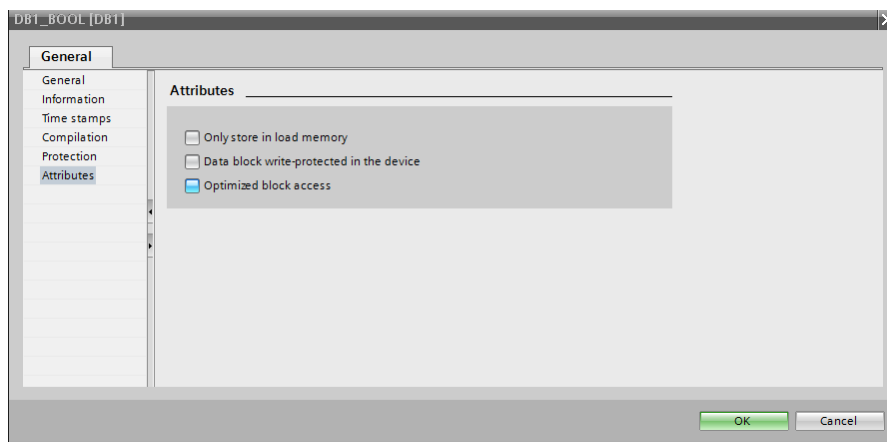
4. In the left pane, click **Protection**.
5. In the **Connection mechanisms** area, select the **Permit access with PUT/GET communication from remote partner** check box.
6. Click **OK**.

### To disable optimized block access:

1. In the TIA software, locate the data block in the left tree.



2. Right-click the data block, and then click **Properties**. The properties dialog box appears.



3. In the left pane, click **Attributes**.
4. Clear the **Optimized block access** check box.
5. Click **OK**.

## Configuring the SIDirect Legacy Object

The SIDirect Legacy object supports "traditional" absolute addressing of PLC items. If you are using symbolic addressing, use the SIDirect Symbolic object.

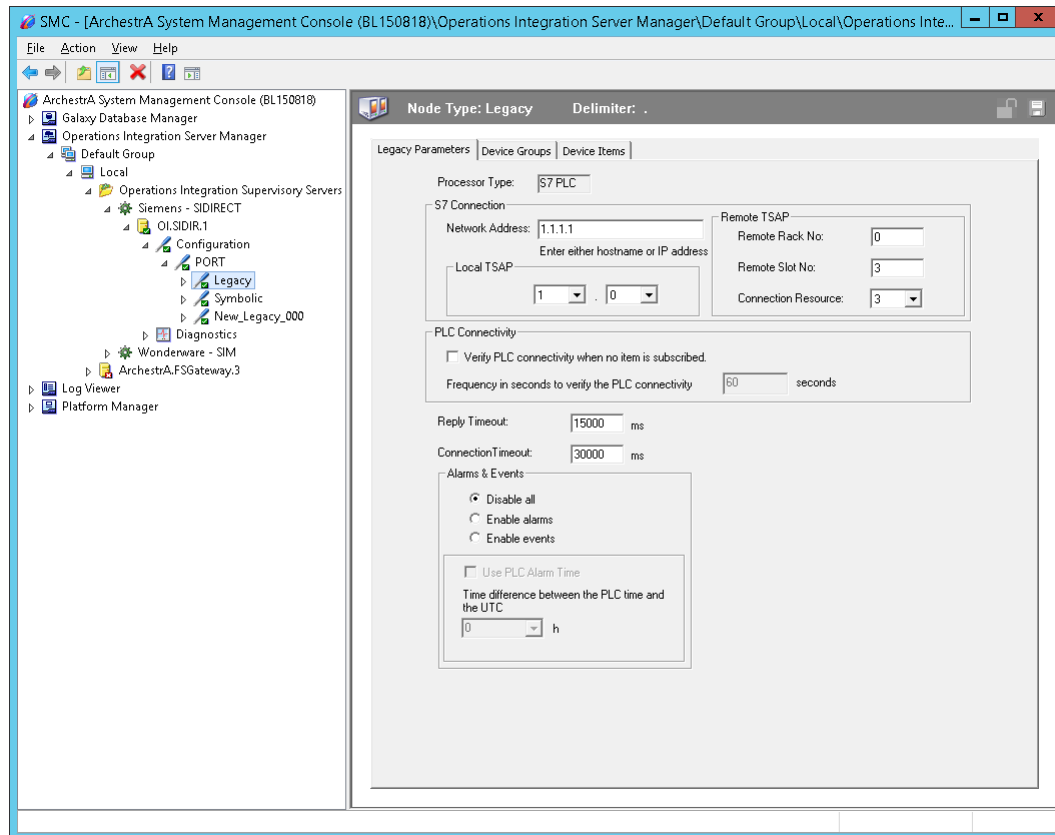
---

**Note:** You will not be able to use the SIDirect Legacy object to access tags in the S7-1500 PLC if the PLC has been configured with a password for **No Access**.

---

### To view or edit the SIDirect Legacy object:

1. Select and right-click on the Legacy object. The object's parameters page appears.



2. You can rename the object as needed.

### To add more Legacy objects:

1. Select and right-click on the PORT object.
2. Select **Add Legacy Object**.
3. A new object named **New\_Legacy\_xxx** is added to the hierarchy, .Rename as needed.

---

**Note:** The Legacy object is preconfigured with three Device Groups and a default network address. The default network address and preconfigured Device Groups are not present in Legacy objects ("New\_Legacy\_000") that you add. Up to 1024 Legacy and Symbolic objects (combined total) can exist in the hierarchy under PORT.

---

## SIDirect Legacy Object Parameters

Edit the elements in the Parameters page as needed. Five of the six elements in the page are configurable.

The screenshot shows the 'Legacy Parameters' configuration window. At the top, there are tabs for 'Legacy Parameters', 'Device Groups', and 'Device Items'. The 'Processor Type' is set to 'S7 PLC'. The 'S7 Connection' section includes a 'Network Address' field with '1.1.1.1' and a note 'Enter either hostname or IP address'. Below it is a 'Local TSAP' section with two dropdown menus showing '1' and '0'. To the right is a 'Remote TSAP' section with three dropdown menus: 'Remote Rack No.' (0), 'Remote Slot No.' (3), and 'Connection Resource' (3). The 'PLC Connectivity' section has a checkbox 'Verify PLC connectivity when no item is subscribed' which is unchecked, and a 'Frequency in seconds to verify the PLC connectivity' field set to '60'. Below that are 'Reply Timeout' (15000 ms) and 'ConnectionTimeout' (30000 ms) fields. The 'Alarms & Events' section has three radio buttons: 'Disable all' (selected), 'Enable alarms', and 'Enable events'. There is also a checkbox 'Use PLC Alarm Time' which is unchecked, and a 'Time difference between the PLC time and the UTC' field set to '0 h'.

- **Processor Type** (not configurable): S7 PLC.
- **S7 Connection:** This has three configurable settings:
  - **Network Address:** The IP address or host name of the remote S7 CP.  
Enter 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 field cannot be blank and the IP address or host name cannot be more than 255 characters.

---

**Note:** The preconfigured Legacy object includes a default network address: "1.1.1.1". Any additional Legacy objects that you add do not contain the default address (network address is blank),

---

- **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 OI Server 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 any one connection, you can configure Alarms, Events, or none. 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.

You can enable this feature to time stamp data in the alarm and event blocks with the timestamps provided in the alarm and event blocks. The OI Server, in this case, will not generate its timestamp for the received data. This feature is only available for data in the alarm and event blocks.
- **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.

## SIDirect Legacy Device Groups and Device Items

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

- For DDE/SuiteLink communications, one or more device-group definitions must exist for each PLC with which the SIDirect OI Server communicates.
- 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 OI Server hierarchy tree has a **Save** button. When you modify the **Parameters**, **Device Groups**, 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.

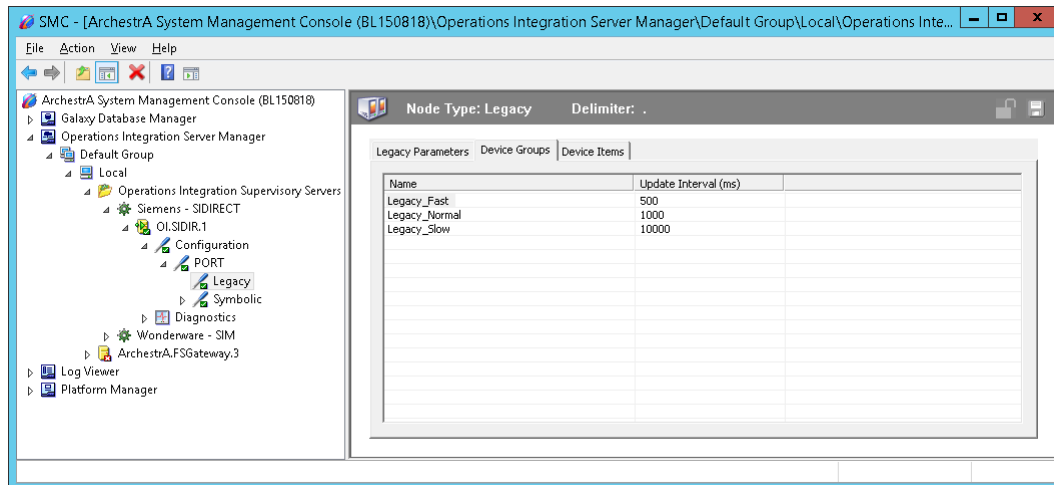
### Editing SIDirect Legacy Object Device Group Parameters

Access the Device Groups dialog box by clicking the Device Groups tab of the SIDirect Legacy Object. In this dialog box, you can:

- Rename device groups
- Add device groups

- Edit device group parameters
- Delete device groups
- Configure a default update interval for objects

**Note:** You are prompted to save the modifications to the configuration set when you select another part of the Server tree hierarchy.



When you install the SIDirect OI Server, it builds a default hierarchy that includes a Legacy object that is preconfigured with three default Device Groups:

- Legacy\_Fast      500 ms update interval
- Legacy\_Normal    1000 ms update interval
- Legacy\_Slow      10000 ms update interval

### To edit update intervals

Update Interval is the frequency in milliseconds that the SIDirect OI Server acquires data from the topics associated with that device group.

- To edit the update interval for an object, double-click its value in the **Update Interval** column and make the edits.

### To configure a default update interval

- To configure a default update interval for the object, right-click in the **Device Groups** box and select the **Config Default Update Interval** option from the menu.

### To 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 not case-sensitive.
3. A default update interval is automatically added. To change the interval, double-click on entry and enter the new update interval (in milliseconds).

### To delete device groups

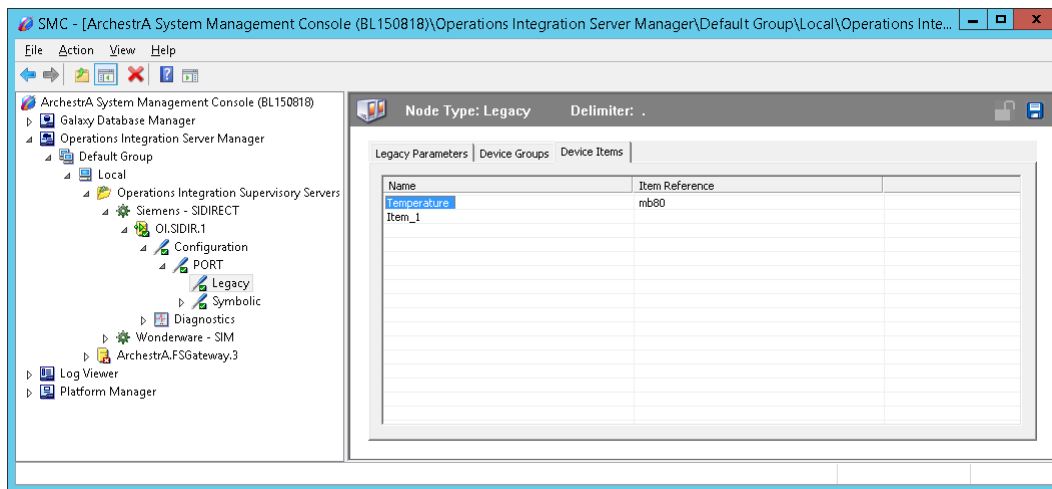
- Right-click on the device group to be deleted from the list and select **Delete**.

## SIDirect Legacy Object Device Item Definitions

The predefined item syntax/name for S7-1500 PLCs 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."

Select the **Device Items** tab of the SIDirect Legacy object to create new, modify, delete, export, or import device-item definitions for an object.

After you configure item names, the OI Server can perform OPC Item browsing. When the OI Server 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 add device items

1. Right-click in the **Device Items** box and click **Add**.
2. Type the item name in the **Name** column, for example, "Temperature."

---

**Note:** When you add a new device item, enter a unique name.

---

3. 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 OI Server 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

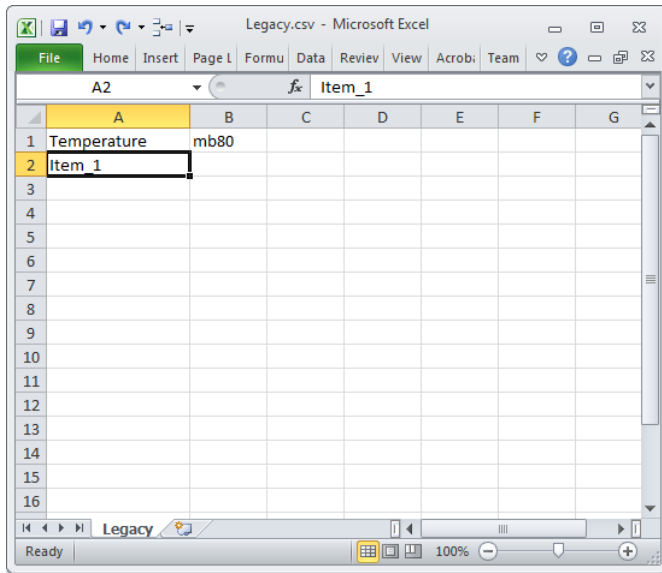
- Right-click in the **Device Items** box and click **Clear All**. All the device items listed are cleared after you confirm their deletion.

## Exporting and Importing SIDirect OI Server Item Data

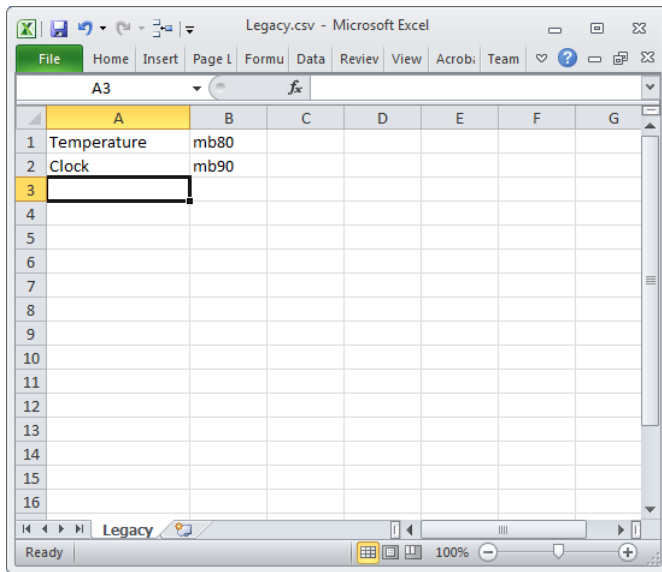
You can export and import the OI Server item data to and from a CSV file after you configure the Device Items. This lets you perform an off-line, large-scale edit on the item data configured for a PLC and import that data back into the PLC configuration.

### To export OI Server 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 "Legacy.csv," within the current-system-configured default directory.
2. Accept the defaults to save the file. The file is saved as Legacy.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.



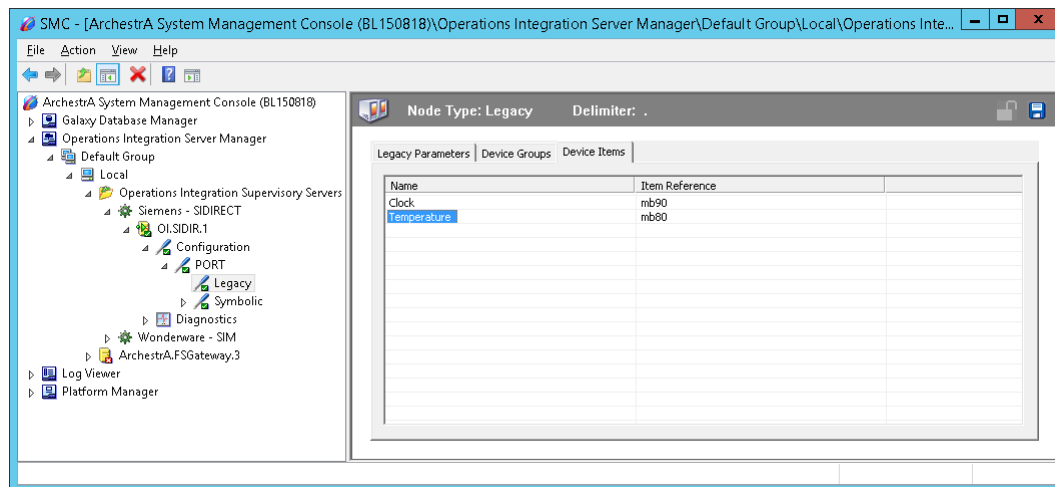
After you are done editing, you are ready to import the edited file into the OI Server Manager.

### To import OI Server 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 **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 OI Server 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 OI Server is running and an OPC Client requests item information, the imported configured items appear under the PLC hierarchy node.

---

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

---

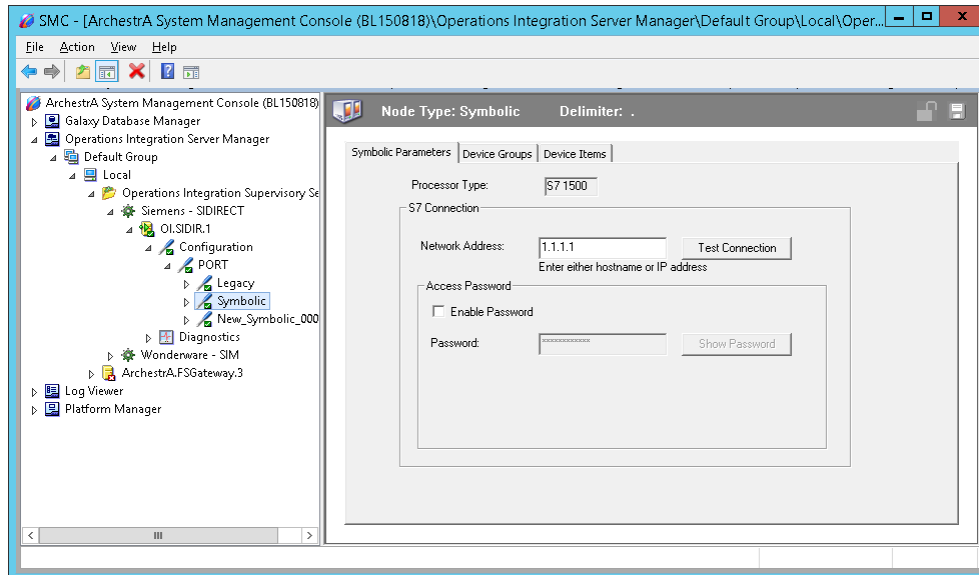
## Configuring the SIDirect Symbolic Object

The SIDirect Symbolic object supports symbolic addressing of PLC items. If you are using absolute addressing, use the SIDirect Legacy object.



### To view or edit the SIDirect Symbolic object:

1. Select and right-click on the Symbolic object. The object's parameters page appears.



2. You can rename the object as needed.

### To add more SIDirect Symbolic objects:

1. Select and right-click on the PORT object.
2. Select **Add Symbolic Object**.
3. A new object named **New\_Symbolic\_xxx** is added to the hierarchy, .Rename as needed.

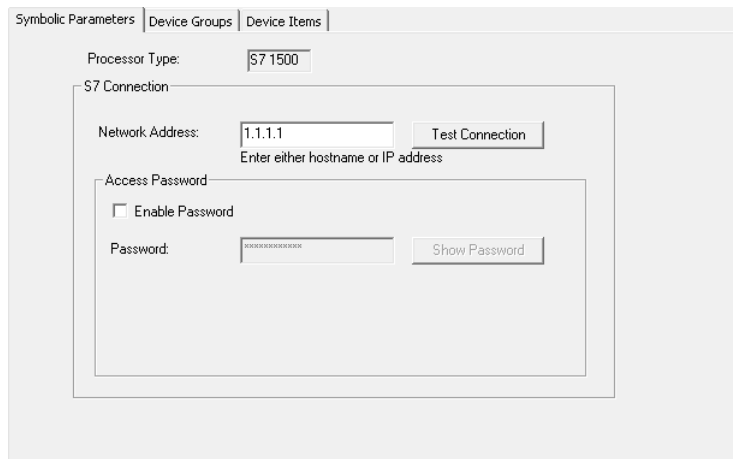
---

**Note:** The SIDirect Symbolic object is preconfigured with three Device Groups and a default network address. The default network address and preconfigured Device Groups are not present in additional Symbolic objects ("New\_Symbolic\_000") that you add. Up to 1024 S7 objects (combined total of Legacy and Symbolic objects) can be added under PORT.

---

## SIDirect Symbolic Object Parameters

Edit the elements in the Parameters page as needed. Configurable elements are network address and password settings.



- **Processor Type** (not configurable): S7 1500.

- **S7 Connection:** This has two configurable settings:
    - **Network Address:** The IP address or host name of the remote S7 1500 PLC.  
Enter the network address (for example, "10.11.12.13"), or a host name if one is defined in the local hostlist. The address field cannot be blank and the IP address or host name cannot be more than 255 characters.  

---

**Note:** The preconfigured SIDIrect Symbolic object includes a default network address: "1.1.1.1". Any additional Symbolic objects that you add do not contain the default address (network address is blank).

---
    - **Enable Password:** You can enable password protection for access to the PLC. Selecting this option activates the password entry fields. You also have the option to display the password characters as you enter them, or to keep them hidden.  

---

**Note:** To help ensure security, passwords for Symbolic node objects can only be set on the local node and cannot be transferred from one Configuration File (.aaCfg) to another. Therefore, if Password is enabled on any Symbolic node object, each must be re-entered after any of the following operations:
      - Copying the Configuration File to another computer
      - Cloning the base server
      - Creating an instance of the base server

---
- 
- Note:** If the S7-1500 PLC has been configured with a password for **No Access**, you will not be able to use the SIDIrect **Legacy** object to access tags in the PLC.
- 

## SIDIrect Symbolic Device Groups and Device Items

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

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

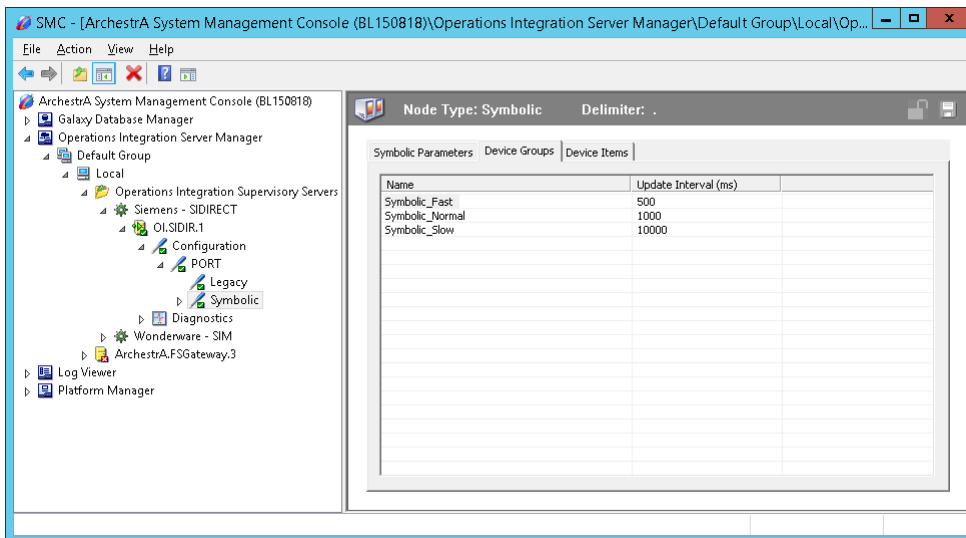
Each configuration view associated with objects in the SIDIrect OI Server hierarchy tree has a **Save** button. When you modify the **Parameters**, **Device Groups**, or the **Device Items** dialog box, click **Save** to implement the new modifications. If you try to open another configuration dialog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

### Editing SIDIrect Symbolic Object Device Group Parameters

Access the Device Groups dialog box by clicking the Device Groups tab of the SIDIrect Symbolic Object. In this dialog box, you can:

- Rename device groups
- Add device groups
- Delete device groups
- Configure a default update interval for objects

**Note:** You are prompted to save the modifications to the configuration set when you select another part of the Server tree hierarchy.



When you install the SIDirect OI Server, it builds a default hierarchy that includes a Symbolic object that is preconfigured with three default Device Groups:

- Symbolic\_Fast                      500 ms update interval
- Symbolic\_Normal                    1000 ms update interval
- Symbolic\_Slow                      10000 ms update interval

### To edit update intervals

Update Interval is the frequency in milliseconds that the SIDirect OI Server acquires data from the topics associated with that device group.

- To edit the update interval for an object, double-click its value in the **Update Interval** column and make the edits.

### To configure a default update interval

- To configure a default update interval for the object, right-click in the **Device Groups** box and select the **Config Default Update Interval** option from the menu. Then, double-click the default time to enter a new default setting, The new default setting will be automatically entered for new Device Groups that you add (update intervals can be changed individually from the default).

### To 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 not case-sensitive.
3. The default update interval is automatically added. To change the interval, double-click on entry and enter the new update interval (in milliseconds).

### To delete device groups

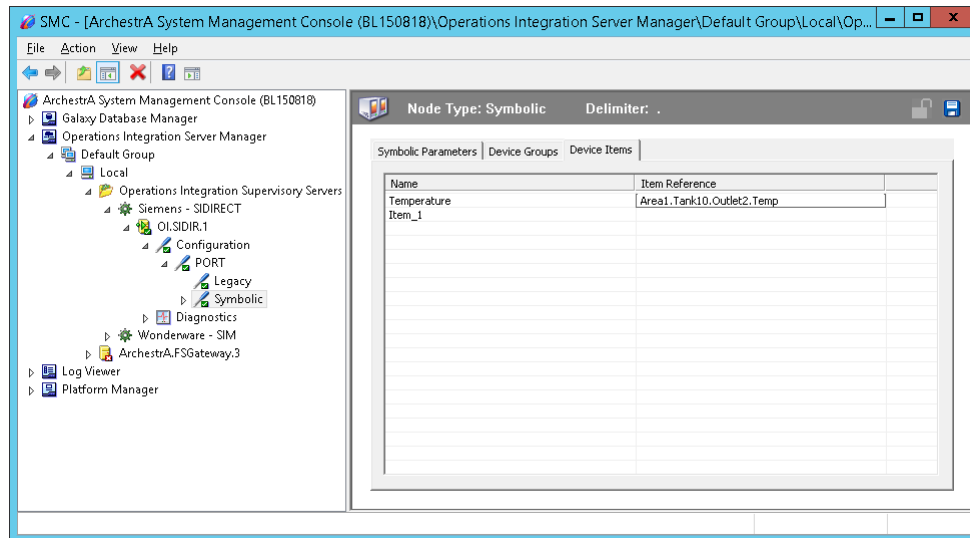
- Right-click on the device group to be deleted from the list and select **Delete**.

## SIDirect Symbolic Object Device Item Definitions

The Device Items tab allows you to provide an alias name for the items that you configure in the PLC (for example, you can define "Temperature" as an alias for "Area1.Tank10.Outlet2.Temp").

Select the **Device Items** tab of the SIDirect Symbolic object to create new, modify, delete, export, or import device-item definitions for an object.

After you configure item names, the OI Server can perform OPC Item browsing. When the OI Server 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 add device items

1. Right-click in the **Device Items** box and click **Add**.
2. Type the item name in the **Name** column, for example, "Temperature."

---

**Note:** When you add a new device item, enter a unique name.

---

3. Double-click the line in the **Item Reference** column and enter the correlated item reference for the name you selected, for example, "Area1.Tank10.Outlet2.Temp".

---

**Note:** If the name and the item reference are the same, it is only necessary to enter a name. The OI Server 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.

## Device Redundancy

The OI Server Manager provides the ability to assign a redundant device for fail-over protection in the event of a device failure.

---

**Note:** Both the SIDirect Legacy and Symbolic objects support redundancy.

---

Two identical devices are expected to be configured in the OI Server Manager having identical item syntax, connected to the same OI Server. For example, you might create two Legacy objects for a redundant pair of S7 PLCs.

---

**Note:** Items cannot be added for updates through the Redundant Device Object if the items do not exist in both controllers.

---

## Run-time Behavior of Redundant Devices

The OI Server will start with the active device. The OI Core will switch to the standby device when the active device fails to communicate. The value of the `$$SYS$Status` will determine the communication failure.

---

**Note:** The value of the `$$SYS$Status` of the standby device must be TRUE in order to switch over to the standby device. Otherwise, there will not be any failover.

---

When `$$SYS$Status` shows a FALSE value at both active and standby devices, the DAS Engine will consider a complete communication failure and mark all the items subscribed to the redundancy device hierarchy with the current time and the appropriate OPC quality. The DAS Engine will activate the slow-poll mechanism to retry the communication to both devices until either one of the Ping Items returns to a good quality and update its `$$SYS$Status` item to TRUE.

When the DAS Engine switches to the standby device, the standby device becomes active and the originally active device becomes the standby.

When the active device becomes the standby device the Ping Item will not be deleted from that the standby device. This will ensure the standby will be able to recover the communication again.

---

**Note:** To allow the failover to function properly, the Ping Item must be a valid PLC item that has not been rejected the server. System items (items beginning with `$$SYS$`) cannot be used as the Ping Item. See *SIDirect Driver Diagnostic Info Items* on page 86 for the list of system items.

---

The OI Server will log any failover activities. All other functionality such as diagnostics, enable/disable, and reset will be performed exactly same as it is performed for any other hierarchy node.

---

**Note:** Unsolicited message configuration is not supported in the Redundant Device Object itself. You can still receive unsolicited messages directly from device groups defined in the regular server hierarchy.

---

This feature allows the OI Server to provide fail over support by providing one node which switches between two other nodes. The Redundant device is configured with a redundancy node which directs itself to one of the two nodes and switches to the other based on lack of communications to a common user-configured controller item. In this manner the Redundant Device Object can be used to direct client requests to the redundant node, which switches between device or communication pathway failure without intervention.

In both stand-alone and redundant configurations, the SIDirect OI Server supports subscription and poking. Block services are not supported.

## Configuring Device Redundancy

The OI Server Manager provides the ability to assign redundant device for fail-over protection in the event of device failure. Two devices must be configured in the same OI Server having identical item syntax.

Primary and secondary devices will be set up in the REDUNDANT\_DEVICE object in the SMC, along with a common item name (ping item) shared by each device to determine device status.

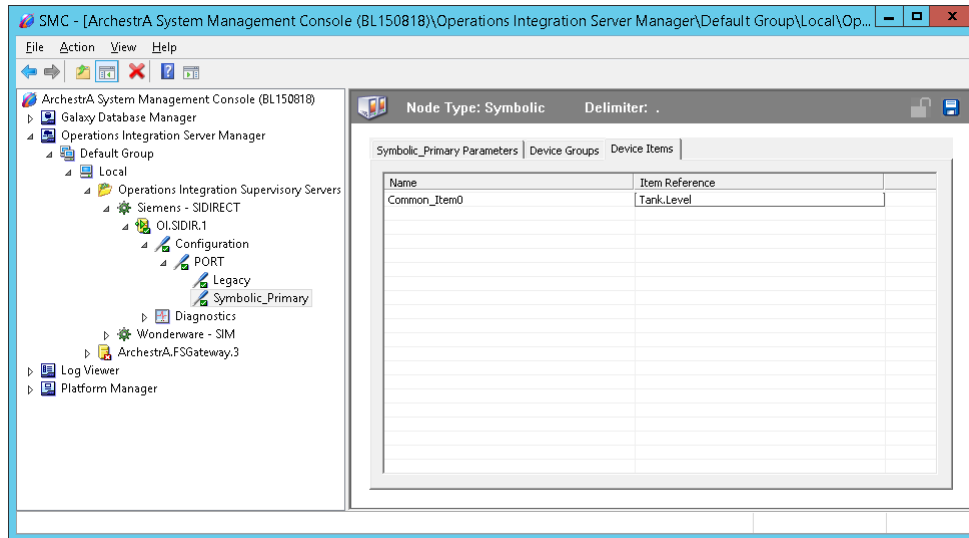
---

**Note:** Unsolicited message configuration is not supported from the device redundant hierarchy.

---

### To setup up a REDUNDANT\_DEVICE from the configuration branch

1. Configure a primary device in the OI Server Manager in the SMC. You must enter a valid network address in the configuration parameters, and add an item reference that can be shared with the the device's secondary partner.

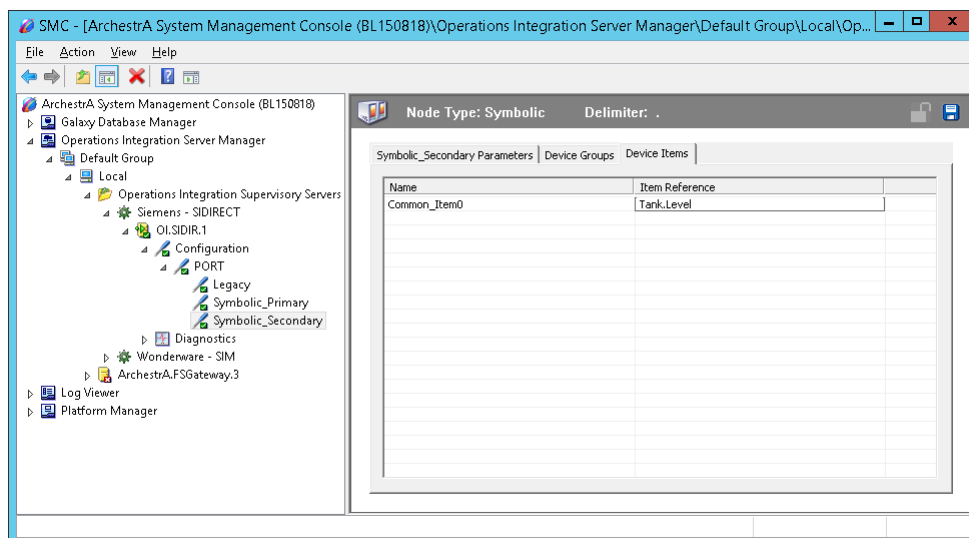


2. Configure a secondary device. Enter a valid network address and add the same item reference that was added for the primary device.

---

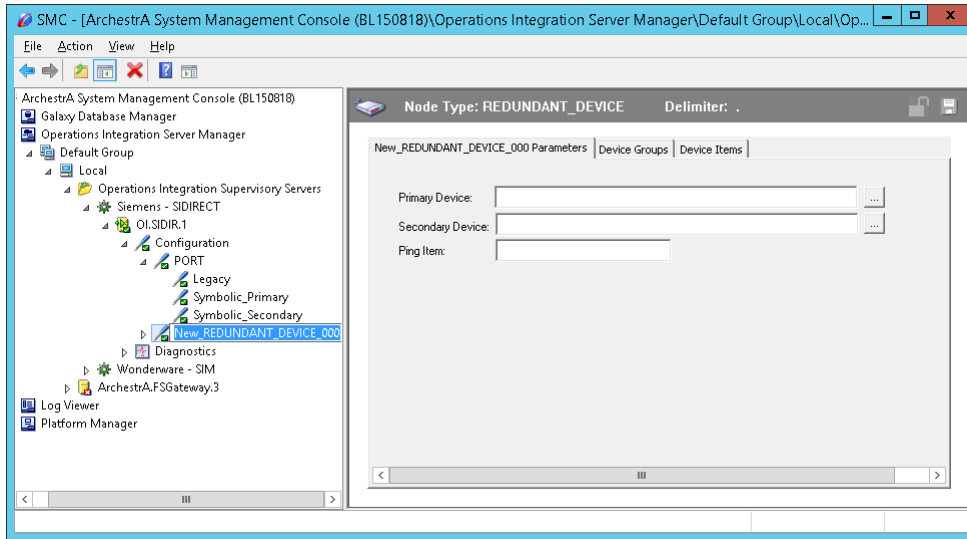
**Important:** You cannot mix Legacy and Symbolic objects when configuring a redundant pair, since these objects use different item syntaxes. Both primary and secondary devices must be the same type.

---

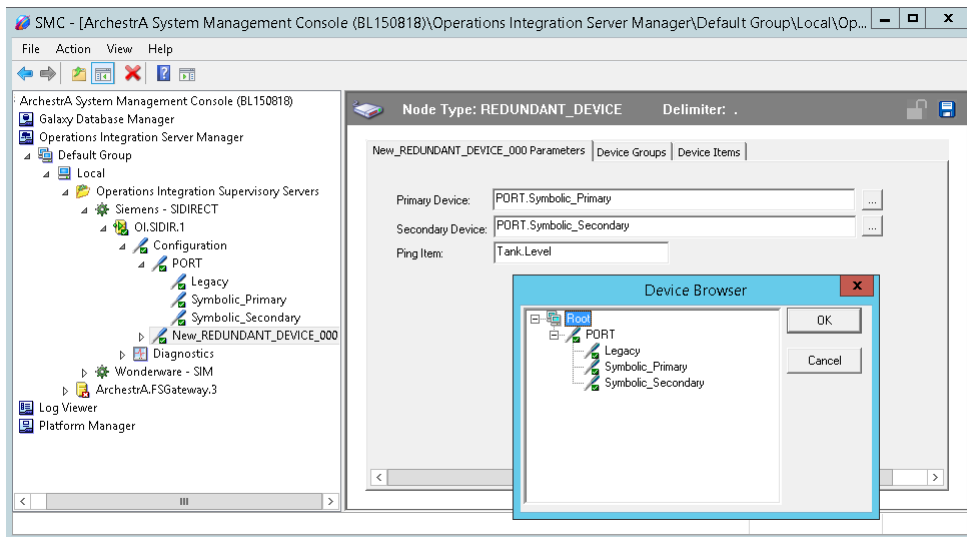


3. Select and right-click on the configuration node under the OI.SIDIR.1 object.

4. Select Add REDUNDANT\_DEVICE Object. An object called New\_REDUNDANT\_DEVICE\_000 is created.



5. Rename the newly created object as appropriate. The New\_REDUNDANT\_DEVICE\_000 configuration view is displayed in the Configuration branch of the hierarchy.
6. Enter or use the device browser to select the primary and secondary devices. Open the device browser by clicking on the ellipses (...) button and expand the device hierarchy that is displayed in the browser.



7. Enter one device item as the Ping Item that can be shared between the primary and secondary devices to determine device status.
8. Save the hierarchy node configuration by clicking on the save icon.

---

**Important:** A Ping item must be specified and be a valid tag in both the primary and secondary controllers to determine the connection status for \$SYS\$Status. The Ping item can be a static item in the device such as a firmware version or processor type. If the Ping item is invalid or does not exist in the controller, the failover operation may not work correctly as the value of \$SYS\$Status may continue to stay as FALSE in the standby device.

---

## SIDirect Scan-Based Message Handling

Wonderware OI Servers 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 OI Server creates its own request and sends that request to the hardware device. The SIDirect OI Server then waits for a response to its request. After the SIDirect OI Server 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 OI Server polls a particular device for a specific piece of information in the device group (topic definition) inside the SIDirect OI Server. 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 can 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.



# CHAPTER 3

## SIDIrect OI Server Reference

### In This Chapter

SIDIrect Legacy Object Reference (Absolute Addressing) .....	33
SIDIrect Symbolic Object Reference (Symbolic Addressing) .....	73
Data Conversion.....	90
Quality Settings .....	91
Item Validation.....	92
Tested Hardware and Firmware Versions.....	93

### SIDIrect Legacy Object Reference (Absolute Addressing)

This section provides reference information specific to the SIDIrect Legacy object. The Legacy object is used for absolute addressing (non-symbolic names).

#### SIDIrect Absolute Naming Convention

The SIDIrect Legacy interface included with the SIDIrect OI Server supports only "traditional" absolute addressing of PLC items. To use absolute addressing:

- Optimized Block Access must be disabled (non-optimized mode).
- GET/PUT must be enabled.

---

**Note:** Symbolic addressing is only supported through the S7\_1500\_Plus interface.

---

In general, the item syntax takes the following format:

```
D[DataBlock#], [DataTypePrefix] [OffsetAddress]<.BitAddress>, <ArraySize>
<DataDisplayFormatSuffix>
```

Example:

DB10, INT2, 8                      Data Block #10 Integer array item of 8 elements starting at 2 bytes offset

DB25, DWord4 BCD                  Data Block #25 Double Word item shown in BCD starting at 4 bytes offset

---

**Note:** The common "D" data block identifier above maybe replaced by a prefix letter such as F (Flag), M (Memory), I (Input), Q (Output), PI (Peripheral Input), PQ (Peripheral Output), T (Timer), C (Counter) for register items from various special Areas in the S7 PLC program.

---

## S7-1500 Item Syntax

The S7-1500 PLCs are a newer series of Siemens S7 PLCs. Its item syntax follows a similar convention as the S7 300/400 series, with some differences. The S7-1500 PLCs support the additional datatypes of LREAL and DTL (Date Time Long).

## S7-1500 Item Naming

The Wonderware SIdirect OI Server 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 S7-1500 PLCs. The ranges specified in those tables vary according to the type of the controller used.

## S7-1500 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 applicable to S7-1500 PLCs. See *Conversions and Suffixes of Items (Absolute Addressing)* on page 70 for suffix definitions.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	D<B,l>d,Xx.y		VT_BOOL	0 or 1
String	D<B,l>d,Sx,v		VT_BSTR	String
	D<B,l>d,STRINGx,v		VT_BSTR	String
S7String	D<B,l>d,S7Sx,w		VT_BSTR	String
	D<B,l>d,S7STRINGx,w		VT_BSTR	String
Byte	D<B,l>d,Bx		VT_UI1	0 to 255
	D<B,l>d,BYTEx		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,l>d,Bx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
	D<B,l>d,BYTEx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
USINT	D<B,l>d,USINTx		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999***
USINT Array	D<B,l>d,USINTx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
Char	D<B,l>d,CHARx		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	D<B,l>d,CHARx,v		VT_ARRAY VT_UI1	-128 to 127 for each element*

<b>Data Format</b>	<b>Item/Point</b>	<b>Suffix</b>	<b>Data Type</b>	<b>Range</b>
SINT	D<B,l>d,SINTx		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999***
SINT Array	D<B,l>dSINTx,v		VT_ARRAY VT_UI1	-128 to 127 for each element*
Word	D<B,l>d,Wn D<B,l>d,WORDn		VT_UI2	0 to 65535
			VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999***
		KT	VT_BSTR	0.0 to 999.3***
		S5T	VT_BSTR	0ms to 2h46m30s0ms***
		TR	VT_R4	0.0 to 9990.0 (s)***
		D	VT_BSTR	1990-1-1 to 2168-12-31
Word Array	D<B,l>d,Wn,v		VT_ARRAY VT_UI2	0 to 65535 for each element*
	D<B,l>d,WORDn,v		VT_ARRAY VT_UI2	0 to 65535 for each element*
UINT	D<B,l>d,UINTn		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999***
		KT	VT_BSTR	0.0 to 999.3***
		S5T	VT_BSTR	0ms to 2h46m30s0ms***
		TR	VT_R4	0.0 to 9990.0 (s)***
		D	VT_BSTR	1990-1-1 to 2168-12-31
UINT Array	D<B,l>d,UINTn,v		VT_ARRAY VT_UI2	0 to 65535 for each element*
Integer	D<B,l>d,INTn		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,l>d,INTn,v		VT_ARRAY VT_UI1	-32768 to 32767 for each element*
Double Word	D<B,l>d,Dm D<B,l>d,DWORDm		VT_UI4	0 to 4294967295**
			VT_UI4	0 to 4294967295**
		BCD	VT_UI4	0 to 999999999***
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648M S to 24D_20H_31M_23S_647M S

Data Format	Item/Point	Suffix	Data Type	Range
Double Word Array	D<B,l>d,Dm,v		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
	D<B,l>d,DWORDm,v		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
UDINT	D<B,l>d,UDINTm		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_648M S to 24D_20H_31M_23S_647M S
UDINT Array	D<B,l>d,UDINTm,v		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
Double Integer	D<B,l>d,DINTm		VT_I4	-2147483648 to 2147483647
		BCD	VT_I4	-99999999 to 99999999***
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648M S to 24D_20H_31M_23S_647M S
Double Integer Array	D<B,l>d,DINTm,v		VT_ARRAY VT_I4	-2147483648 to 2147483647 for each element*
Real	D<B,l>d,REALm		VT_R4	+/-1.2e-38 to +/-3.4e+38
Real Array	D<B,l>d,REALm,v		VT_ARRAY VT_R4	+/-1.2e-38 to +/-3.4e+38 for each element*

\*: For DDE/SuiteLink, the item value is the Hex ASCII representation of the complete array. The result is one string containing all the elements of the array in the Hex ASCII 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, 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.
- m is the start address of 4-byte data/4-byte data arrays.
- v is the length of data in elements (an item in an array), 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).

\*\*\*: The OI Server provides suffixes (S5T, TR, DT, KT, and BCD) that are used to interpret raw bytes as having a special data format. Review the PLC processor documentation, as not all processors have ladder logic internally to interpret those data formats the same way (for example, the S7-1200 PLC).

---

**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 OI Server processes a write (**POKE**) to a Data Block.

---

For example:

DB5,DTL5 TREAL

## Flag Bytes

The following table summarizes data format, item or point, suffix, data type, and range for Flag Bytes. See *Conversions and Suffixes of Items (Absolute Addressing)* on page 70 for suffix definitions.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	FXx.y		VT_BOOL	0 or 1
	MXx.y		VT_BOOL	0 or 1
String	FSx,v		VT_BSTR	String
	MSx,v		VT_BSTR	String
	FSTRINGx,v		VT_BSTR	String
	MSTRINGx,v		VT_BSTR	String
Byte	FBx		VT_UI1	0 to 255
	MBx		VT_UI1	0 to 255
	FBYTEx		VT_UI1	0 to 255
	MBYTEx		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	FBx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
	MBx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
	FBYTEx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
	MBYTEx,v		VT_ARRAY VT_UI1	0 to 255 for each element*

<b>Data Format</b>	<b>Item/Point</b>	<b>Suffix</b>	<b>Data Type</b>	<b>Range</b>
USINT	D<B,l>d,USINTx		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999***
USINT Array	D<B,l>d,USINTx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
Char	FCHARx MCHARx		VT_I1	-128 to 127
			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	FCHARx,v MCHARx,v		VT_ARRAY VT_I1	-128 to 127 for each element*
			VT_ARRAY VT_I1	-128 to 127 for each element*
SINT	D<B,l>d,SINTx		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999***
SINT Array	D<B,l>dSINTx,v		VT_ARRAY VT_UI1	-128 to 127 for each element*
Word	FWn MWn FWORDn MWORDn		VT_UI2	0 to 65535
			VT_UI2	0 to 65535
			VT_UI2	0 to 65535
			VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30ms
		TR	VT_R4	0.0 to 9990.0 (s)
	D	VT_BSTR	1990-1-1 to 2168-12-31	
Word Array	FWn,v MWn,v FWORDn,v MWORDn,v		VT_ARRAY VT_UI2	0 to 65535 for each element*
			VT_ARRAY VT_UI2	0 to 65535 for each element*
			VT_ARRAY VT_UI2	0 to 65535 for each element*
			VT_ARRAY VT_UI2	0 to 65535 for each element*
				0 to 65535 for each element*

<b>Data Format</b>	<b>Item/Point</b>	<b>Suffix</b>	<b>Data Type</b>	<b>Range</b>
UINT	D<B,l>d,UINT <i>n</i>		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999***
		KT	VT_BSTR	0.0 to 999.3***
		S5T	VT_BSTR	0ms to 2h46m30s0ms***
		TR	VT_R4	0.0 to 9990.0 (s)***
		D	VT_BSTR	1990-1-1 to 2168-12-31
UINT Array	D<B,l>d,UINT <i>n,v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element*
Integer	FINT <i>n</i> MINT <i>n</i>		VT_I2	-32768 to 32767
			VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	FINT <i>n,v</i>		VT_ARRAY VT_I2	-32768 to 32767 for each element*
	MINT <i>n,v</i>		VT_ARRAY VT_I2	-32768 to 32767 for each element*
Double Word	FD <i>m</i>		VT_UI4	0 to 4294967295**
	MD <i>m</i>		VT_UI4	0 to 4294967295**
	FDWORD <i>m</i>		VT_UI4	0 to 4294967295**
			VT_UI4	0 to 4294967295**
	MDWORD <i>m</i>	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_648M S to 24D_20H_31M_23S_647M S
Double Word Array	FD <i>m,v</i>		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
	MD <i>m,v</i>		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
	FDWORD <i>m,v</i>		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
	MDWORD <i>m,v</i>		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
UDINT	D<B,l>d,UDINT <i>m</i>		VT_UI4	0 to 4294967295**
		BCD	VT_UI4	0 to 999999999***
		TOD	VT_BSTR	0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648M S to

Data Format	Item/Point	Suffix	Data Type	Range
				24D_20H_31M_23S_647M S
UDINT Array	D<B,l>d,UDINT <i>m,v</i>		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
Double Integer	FDINT <i>m</i> MDINT <i>m</i>		VT_14	-2147483648 to 2147483647
		BCD	VT_14	-2147483648 to 2147483647
		TOD	VT_BSTR	-9999999 to 9999999
		T	VT_BSTR	0:00:00.000 to 23:59:59.999
Double Integer Array	FDINT <i>m,v</i>		VT_ARRAY VT_I4	-2147483648 to 2147483647 for each element*
	MDINT <i>m,v</i>		VT_ARRAY VT_I4	-2147483648 to 2147483647 for each element*
Real	FREAL <i>m</i>		VT_R4	+/-1.2e-38 to +/-3.4e+38
	MREAL <i>m</i>		VT_R4	+/-1.2e-38 to +/-3.4e+38
Real Array	FREAL <i>m,v</i>		VT_ARRAY VT_R4	+/-1.2e-38 to +/-3.4e+38 for each element*
	MREAL <i>m,v</i>		VT_ARRAY VT_R4	+/-1.2e-38 to +/-3.4e+38 for each element*

\*: For DDE/SuiteLink, the item value is the Hex ASCII representation of the complete array. The result is one string containing all the elements of the array in the Hex ASCII 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.



Where:

- 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 elements (an item in an array), with a range from 1 to (net PDU data size/type size - header information).

\*\*\*: The OI Server provides suffixes (S5T, TR, DT, KT, and BCD) that are used to interpret raw bytes as having a special data format. Review the PLC processor documentation, as not all processors have ladder logic internally to interpret those data formats the same way.

**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 OI Server 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. See *Conversions and Suffixes of Items (Absolute Addressing)* on page 70 for suffix definitions.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	Ix.y		VT_BOOL	0 or 1
	Ex.y		VT_BOOL	0 or 1
	IXx.y		VT_BOOL	0 or 1
	EXx.y		VT_BOOL	0 or 1
String	ISx,v		VT_BSTR	String
	ESx,v		VT_BSTR	String
	ISTRINGx,v		VT_BSTR	String
	ESTRINGx,v		VT_BSTR	String
Byte	IBx		VT_UI1	0 to 255
	EBx		VT_UI1	0 to 255
	IBYTEx		VT_UI1	0 to 255
	EBYTEx		VT_UI1	0 to 255
			DT	VT_BSTR
Byte Array	IBx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
	EBx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
	IBYTEx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
	EBYTEx,v		VT_ARRAY VT_UI1	0 to 255 for each element*

<b>Data Format</b>	<b>Item/Point</b>	<b>Suffix</b>	<b>Data Type</b>	<b>Range</b>
USINT	D<B,l>d,USINTx		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999***
USINT Array	D<B,l>d,USINTx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
Char	ICHARx ECHARx		VT_I1	-128 to 127
			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	ICHARx,v ECHARx,v		VT_ARRAY VT_I1	-128 to 127 for each element*
			VT_ARRAY VT_I1	-128 to 127 for each element*
SINT	D<B,l>d,SINTx		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999***
SINT Array	D<B,l>dSINTx,v		VT_ARRAY VT_UI1	-128 to 127 for each element*
Word	IWn EWn IWORDn EWORDn		VT_UI2	0 to 65535
			VT_UI2	0 to 65535
			VT_UI2	0 to 65535
			VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s0ms
		TR	VT_R4	0.0 to 9990.0 (s)
	D	VT_BSTR	1990-1-1 to 2168-12-31	
Word Array	IWn,v EWn,v IWORDn,v EWORDn,v		VT_ARRAY VT_UI2	0 to 65535 for each element*
			VT_ARRAY VT_UI2	0 to 65535 for each element*
			VT_ARRAY VT_UI2	0 to 65535 for each element*
			VT_ARRAY VT_UI2	0 to 65535 for each element*
				0 to 65535 for each element*

<b>Data Format</b>	<b>Item/Point</b>	<b>Suffix</b>	<b>Data Type</b>	<b>Range</b>	
UINT	D<B,l>d,UINT <i>n</i>		VT_UI2	0 to 65535	
			BCD	VT_UI2	0 to 9999***
			KT	VT_BSTR	0.0 to 999.3***
			S5T	VT_BSTR	0ms to 2h46m30s0ms***
			TR	VT_R4	0.0 to 9990.0 (s)***
			D	VT_BSTR	1990-1-1 to 2168-12-31
UINT Array	D<B,l>d,UINT <i>n,v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element*	
Integer	IINT <i>n</i> EINT <i>n</i>		VT_I2	-32768 to 32767	
			VT_I2	-32768 to 32767	
		BCD	VT_I2	-999 to 999	
		D	VT_BSTR	1990-1-1 to 2168-12-31	
Integer Array	IINT <i>n,v</i> EINT <i>n,v</i>		VT_ARRAY VT_I2	-32768 to 32767 for each element*	
			VT_ARRAY VT_I2	-32768 to 32767 for each element*	
Double Word	ID <i>m</i> ED <i>m</i> IDWORD <i>m</i> EDWORD <i>m</i>		VT_UI4	0 to 4294967295**	
			VT_UI4	0 to 4294967295**	
			VT_UI4	0 to 4294967295**	
			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_648M S to 24D_20H_31M_23S_647M S	
Double Word Array	ID <i>m,v</i> ED <i>m,v</i> IDWORD <i>m,v</i> EDWORD <i>m,v</i>		VT_ARRAY VT_UI4	0 to 4294967295 for each element*	
			VT_ARRAY VT_UI4	0 to 4294967295 for each element	
			VT_ARRAY VT_UI4	0 to 4294967295 for each element*	
			VT_ARRAY VT_UI4	0 to 4294967295 for each element	
UDINT	D<B,l>d,UDINT <i>m</i>		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_648M S to	

Data Format	Item/Point	Suffix	Data Type	Range
				24D_20H_31M_23S_647MS
UDINT Array	D<B,l>d,UDINT <i>m,v</i>		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
Double Integer	IDINT <i>m</i>		VT_I4	-2147483648 to 2147483647
	EDINT <i>m</i>		VT_I4	-2147483648 to 2147483647
		BCD	VT_I4	-2147483648 to 2147483647
		TOD	VT_BSTR	-9999999 to 9999999
		T	VT_BSTR	0:00:00.000 to 23:59:59.999
				-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS
Double Integer Array	IDINT <i>m,v</i>		VT_ARRAY VT_I4	-2147483648 to 2147483647 for each element*
	EDINT <i>m,v</i>		VT_ARRAY VT_I4	-2147483648 to 2147483647 for each element*
Real	IREAL <i>m</i>		VT_R4	+/-1.2e-38 to +/-3.4e+38
	EREAL <i>m</i>		VT_R4	+/-1.2e-38 to +/-3.4e+38
Real Array	IREAL <i>m,v</i>		VT_ARRAY VT_R4	+/-1.2e-38 to +/-3.4e+38 for each element*
	EREAL <i>m,v</i>		VT_ARRAY VT_R4	+/-1.2e-38 to +/-3.4e+38 for each element

\*: For DDE/SuiteLink, the item value is the Hex ASCII representation of the complete array. The result is one string containing all the elements of the array in the Hex ASCII 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.

Where:

- 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 elements (an item in an array), with a range from 1 to (net PDU data size/type size - header information).

\*\*\*: The OI Server provides suffixes (S5T, TR, DT, KT, and BCD) that are used to interpret raw bytes as having a special data format. Review the PLC processor documentation, as not all processors have ladder logic internally to interpret those data formats the same way.

---

**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 OI Server 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. See *Conversions and Suffixes of Items (Absolute Addressing)* on page 70 for suffix definitions.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	Ox.y		VT_BOOL	0 or 1
	Ax.y		VT_BOOL	0 or 1
	Qx.y		VT_BOOL	0 or 1
	OXx.y		VT_BOOL	0 or 1
	AXx.y		VT_BOOL	0 or 1
	QXx.y		VT_BOOL	0 or 1
String	OSx,v		VT_BSTR	String
	ASx,v		VT_BSTR	String
	QSx,v		VT_BSTR	String
	OSTRINGx,v		VT_BSTR	String
	ASTRINGx,v		VT_BSTR	String
	QSTRINGx,v		VT_BSTR	String
Byte	OBx		VT_UI1	0 to 255
	ABx		VT_UI1	0 to 255
	QBx		VT_UI1	0 to 255
	OBYTEx		VT_UI1	0 to 255
	ABYTEx		VT_UI1	0 to 255
	QBYTEx		VT_UI1	0 to 255
			DT	VT_BSTR

Data Format	Item/Point	Suffix	Data Type	Range
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*
USINT	D<B,I> $d$ ,USINT $x$		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999***
USINT Array	D<B,I> $d$ ,USINT $x,v$		VT_ARRAY VT_UI1	0 to 255 for each element*
Char	OCHAR $x$		VT_I1	-128 to 127
			VT_I1	-128 to 127
			VT_I1	-128 to 127
	QCHAR $x$	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*
SINT	D<B,I> $d$ ,SINT $x$		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999***
SINT Array	D<B,I> $d$ SINT $x,v$		VT_ARRAY VT_UI1	-128 to 127 for each element*

Data Format	Item/Point	Suffix	Data Type	Range	
Word	<i>OWn</i>		VT_UI2	0 to 65535	
	<i>AWn</i>		VT_UI2	0 to 65535	
	<i>QWn</i>		VT_UI2	0 to 65535	
	<i>OWORDn</i>		VT_UI2	0 to 65535	
	<i>AWORDn</i>		VT_UI2	0 to 65535	
	<i>QWORDn</i>		VT_UI2	0 to 65535	
			BCD	VT_UI2	0 to 9999
			KT	VT_BSTR	0.0 to 999.3
			S5T	VT_BSTR	0ms to 2h46m30s0ms
			TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31	
Word Array	<i>OWn,v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element*	
	<i>AWn,v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element	
	<i>QWn,v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element	
	<i>OWORDn,v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element	
	<i>AWORDn,v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element*	
	<i>QWORDn,v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element	
UINT	<i>D&lt;B,I&gt;d,UINTn</i>		VT_UI2	0 to 65535	
		BCD	VT_UI2	0 to 9999***	
		KT	VT_BSTR	0.0 to 999.3***	
		S5T	VT_BSTR	0ms to 2h46m30s0ms***	
		TR	VT_R4	0.0 to 9990.0 (s)***	
		D	VT_BSTR	1990-1-1 to 2168-12-31	
UINT Array	<i>D&lt;B,I&gt;d,UINTn,v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element*	
Integer	<i>OINTn</i>		VT_I2	-32768 to 32767	
	<i>AINTn</i>		VT_I2	-32768 to 32768	
	<i>QINTn</i>		VT_I2	-32768 to 32768	
		BCD		-999 to 999	
		D	VT_BSTR	1990-1-1 to 2168-12-31	

Data Format	Item/Point	Suffix	Data Type	Range	
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_648M S to 24D_20H_31M_23S_647M S
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*	
UDINT	D<B,I> $d$ ,UDINT $m$		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_648M S to 24D_20H_31M_23S_647M S	
UDINT Array	D<B,I> $d$ ,UDINT $m,v$		VT_ARRAY VT_UI4	0 to 4294967295 for each element*	



Data Format	Item/Point	Suffix	Data Type	Range	
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	+/-1.2e-38 to +/-3.4e+38	
	AREAL $m$				
	QREAL $m$				
Real Array	OREAL $m,v$		VT_ARRAY VT_R4	+/-1.2e-38 to +/-3.4e+38 for each element*	
	AREAL $m,v$				
	QREAL $m,v$				

\*: For DDE/SuiteLink, the item value is the Hex ASCII representation of the complete array. The result is one string containing all the elements of the array in the Hex ASCII 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 elements (an item in an array), with a range from 1 to (net PDU data size/type size - header information).

\*\*\*: The OI Server provides suffixes (S5T, TR, DT, KT, and BCD) that are used to interpret raw bytes as having a special data format. Review the PLC processor documentation, as not all processors have ladder logic internally to interpret those data formats the same way.

**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 OI Server 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. See *Conversions and Suffixes of Items (Absolute Addressing)* on page 70 for suffix definitions.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	PIx.y		VT_BOOL	0 or 1
	PEx.y		VT_BOOL	0 or 1
	PIXx.y		VT_BOOL	0 or 1
	PEXx.y		VT_BOOL	0 or 1
String	PISx,v		VT_BSTR	String
	PESx,v		VT_BSTR	String
	PISTRINGx,v		VT_BSTR	String
	PESTRINGx,v		VT_BSTR	String
Byte	PIBx		VT_UI1	0 to 255
	PEBx		VT_UI1	0 to 255
	PIBYTEx		VT_UI1	0 to 255
	PEBYTEx		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	PIBx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
	PEBx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
	PIBYTEx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
	PEBYTEx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
USINT	D<B,l>d,USINTx		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999***
USINT Array	D<B,l>d,USINTx,v		VT_ARRAY VT_UI1	0 to 255 for each element*
Char	PICHARx		VT_I1	-128 to 127
	PECHARx		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	PICHARx,v		VT_ARRAY VT_I1	-128 to 127 for each

<b>Data Format</b>	<b>Item/Point</b>	<b>Suffix</b>	<b>Data Type</b>	<b>Range</b>	
	PECHAR <sub>x,v</sub>		VT_ARRAY VT_I1	element* -128 to 127 for each element*	
SINT	D<B,l>d,SINT <sub>x</sub>		VT_I1	-128 to 127	
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999***	
SINT Array	D<B,l>dSINT <sub>x,v</sub>		VT_ARRAY VT_UI1	-128 to 127 for each element*	
Word	PIW <sub>n</sub> PEW <sub>n</sub> PIWORD <sub>n</sub> PEWORD <sub>n</sub>		VT_UI2	0 to 65535	
			VT_UI2	0 to 65535	
			VT_UI2	0 to 65535	
			VT_UI2	0 to 65535	
		BCD	VT_UI2	0 to 9999	
		KT	VT_BSTR	0.0 to 999.3	
		S5T	VT_BSTR	0ms to 2h46m30s0ms	
		TR	VT_R4	0.0 to 9990.0 (s)	
	D	VT_BSTR	1990-1-1 to 2168-12-31		
Word Array	PIW <sub>n,v</sub> PEW <sub>n,v</sub> PIWORD <sub>n,v</sub> PEWORD <sub>n,v</sub>		VT_ARRAY VT_UI2	0 to 65535 for each element*	
			VT_ARRAY VT_UI2	0 to 65535 for each element*	
			VT_ARRAY VT_UI2	0 to 65535 for each element*	
			VT_ARRAY VT_UI2	0 to 65535 for each element*	
UINT	D<B,l>d,UINT <sub>n</sub>		VT_UI2	0 to 65535	
		BCD	VT_UI2	0 to 9999***	
		KT	VT_BSTR	0.0 to 999.3***	
		S5T	VT_BSTR	0ms to 2h46m30s0ms***	
		TR	VT_R4	0.0 to 9990.0 (s)***	
			D	VT_BSTR	1990-1-1 to 2168-12-31
UINT Array	D<B,l>d,UINT <sub>n,v</sub>		VT_ARRAY VT_UI2	0 to 65535 for each element*	
Integer	PIINT <sub>n</sub> PEINT <sub>n</sub>		VT_I2	-32768 to 32767	
			VT_I2	-32768 to 32767	
		BCD	VT_I2	-999 to 999	
		D	VT_BSTR	1990-1-1 to 2168-12-31	

Data Format	Item/Point	Suffix	Data Type	Range	
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*	
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_648M S to 24D_20H_31M_23S_647M S
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	
UDINT	D<B,l>d,UDINT $m$		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_648M S to 24D_20H_31M_23S_647M S	
UDINT Array	D<B,l>d,UDINT $m,v$		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	-2147483648 to 2147483647
			TOD	VT_BSTR	-9999999 to 9999999
			T	VT_BSTR	0:00:00.000 to 23:59:59.999 -24D_20H_31M_23S_648M S to 24D_20H_31M_23S_647M S

Data Format	Item/Point	Suffix	Data Type	Range
Double Integer Array	PIDINT $m,v$ PEDINT $m,v$		VT_ARRAY VT_I4	-2147483648 to 2147483647 for each element*
Real	PIREAL $m$ PEREAL $m$		VT_R4 VT_R4	+/-1.2e-38 to +/-3.4e+38 +/-1.2e-38 to +/-3.4e+38
Real Array	PIREAL $m,v$ PEREAL $m,v$		VT_ARRAY VT_R4 VT_ARRAY VT_R4	+/-1.2e-38 to +/-3.4e+38 for each element* +/-1.2e-38 to +/-3.4e+38 for each element

\*: For DDE/SuiteLink, the item value is the Hex ASCII representation of the complete array. The result is one string containing all the elements of the array in the Hex ASCII 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 elements (an item in an array), with a range from 1 to (net PDU data size/type size - header information).

\*\*\*: The SIDirect OI Server provides suffixes (S5T, TR, DT, KT, and BCD) that are used to interpret raw bytes as having a special data format. Review the PLC processor OI Server documentation, as not all processors have ladder logic internally to interpret those data formats the same way.

**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 OI Server 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. See *Conversions and Suffixes of Items (Absolute Addressing)* on page 70 for suffix definitions.

<b>Data Format</b>	<b>Item/Point</b>	<b>Suffix</b>	<b>Data Type</b>	<b>Range</b>
Bit	POx.y		VT_BOOL	0 or 1
	PAx.y		VT_BOOL	0 or 1
	PQx.y		VT_BOOL	0 or 1
	POXx.y		VT_BOOL	0 or 1
	PAXx.y		VT_BOOL	0 or 1
	PQXx.y		VT_BOOL	0 or 1
String	POSx.v		VT_BSTR	String
	PASx.v		VT_BSTR	String
	PQSx.v		VT_BSTR	String
	POSTRINGx.v		VT_BSTR	String
	PASTRINGx.v		VT_BSTR	String
	PQSTRINGx.v		VT_BSTR	String
Byte	POBx		VT_UI1	0 to 255
	PABx		VT_UI1	0 to 255
	PQBx		VT_UI1	0 to 255
	POBYTEx		VT_UI1	0 to 255
	PABYTEx		VT_UI1	0 to 255
	PQBYTEx		VT_UI1	0 to 255
			DT	VT_BSTR
Byte Array	POBx.v		VT_ARRAY VT_UI1	0 to 255 for each element*
	PABx.v		VT_ARRAY VT_UI1	0 to 255 for each element*
	PQBx.v		VT_ARRAY VT_UI1	0 to 255 for each element*
	POBYTEx.v		VT_ARRAY VT_UI1	0 to 255 for each element*
	PABYTEx.v		VT_ARRAY VT_UI1	0 to 255 for each element*
	PQBYTEx.v		VT_ARRAY VT_UI1	0 to 255 for each element*
USINT	D<B,l>d,USINTx		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999** *
USINT Array	D<B,l>d,USINTx,v		VT_ARRAY VT_UI1	0 to 255 for each element*

<b>Data Format</b>	<b>Item/Point</b>	<b>Suffix</b>	<b>Data Type</b>	<b>Range</b>
Char	PACHARx		VT_I1	-128 to 127
	POCHARx		VT_I1	-128 to 127
	PQCHARx		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	POCHARx.v		VT_ARRAY VT_I1	-128 to 127 for each element*
	PACHARx.v		VT_ARRAY VT_I1	-128 to 127 for each element*
	PQCHARx.v		VT_ARRAY VT_I1	-128 to 127 for each element*
SINT	D<B,l>d,SINTx		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999** *
SINT Array	D<B,l>dSINTx,v		VT_ARRAY VT_UI1	-128 to 127 for each element*
Word	POWn		VT_UI2	0 to 65535
	PAWn		VT_UI2	0 to 65535
	PQWn		VT_UI2	0 to 65535
	POWORDn		VT_UI2	0 to 65535
	PAWORDn		VT_UI2	0 to 65535
	PQWORDn		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s0ms
		TR	VT_R4	0.0 to 9990.0 (s)
	D	VT_BSTR	1990-1-1 to 2168-12-31	
Word Array	POWn,v		VT_ARRAY VT_UI2	0 to 65535 for each element*
	PAWn,v		VT_ARRAY VT_UI2	0 to 65535 for each element*
	PQWn,v		VT_ARRAY VT_UI2	0 to 65535 for each element*
	POWORDn,v		VT_ARRAY VT_UI2	0 to 65535 for each element*
	PAWORDn,v		VT_ARRAY VT_UI2	0 to 65535 for each element*
	PQWORDn,v		VT_ARRAY VT_UI2	0 to 65535 for each element*

<b>Data Format</b>	<b>Item/Point</b>	<b>Suffix</b>	<b>Data Type</b>	<b>Range</b>
UINT	D<B,l>d,UINT <i>n</i>		VT_UI2	0 to 65535
			BCD VT_UI2	0 to 9999***
			KT VT_BSTR	0.0 to 999.3***
			S5T VT_BSTR	0ms to 2h46m30s0ms***
			TR VT_R4	0.0 to 9990.0 (s)***
			D VT_BSTR	1990-1-1 to 2168-12-31
UINT Array	D<B,l>d,UINT <i>n,v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element*
Integer	POINT <i>n</i> PAINT <i>n</i> PQINT <i>n</i>		VT_I2	-32768 to 32767
			VT_I2	-32768 to 32767
			VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	POINT <i>n,v</i> PAINT <i>n,v</i> PQINT <i>n,v</i>		VT_ARRAY VT_I2	-32768 to 32767 for each element*
			VT_ARRAY VT_I2	-32768 to 32767 for each element*
			VT_ARRAY VT_I2	-32768 to 32767 for each element*
				-32768 to 32767 for each element*
Double Word	POD <i>m</i> PAD <i>m</i> PQD <i>m</i> PODWORD <i>m</i> PADWORD <i>m</i> PQDWORD <i>m</i>		VT_UI4	0 to 4294967295**
			VT_UI4	0 to 4294967295**
			VT_UI4	0 to 4294967295**
			VT_UI4	0 to 4294967295**
			VT_UI4	0 to 4294967295**
			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 <i>m,v</i> PAD <i>m,v</i> PQD <i>m,v</i>		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
			VT_ARRAY VT_UI4	0 to 4294967295 for each element*
			VT_ARRAY VT_UI4	0 to 4294967295 for each element*



<b>Data Format</b>	<b>Item/Point</b>	<b>Suffix</b>	<b>Data Type</b>	<b>Range</b>
UDINT	D<B,l>d,UDINT <i>m</i>		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_647M S
UDINT Array	D<B,l>d,UDINT <i>m,v</i>		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
Double Integer	PODINT <i>m</i>		VT_I4	-2147483648 to 2147483647
	PADINT <i>m</i>		VT_I4	-2147483648 to 2147483647
	PQDINT <i>m</i>		VT_I4	-2147483648 to 2147483647
		BCD	VT_I4	-2147483648 to 2147483647
		TOD	VT_BSTR	-9999999 to 9999999  0:00:00.000 to 23:59:59.999
		T	VT_BSTR	-24D_20H_31M_23S_648 MS to 24D_20H_31M_23S_647M S
Double Integer Array	PODINT <i>m,v</i>		VT_ARRAY VT_I4	-2147483648 to 2147483647 for each element*
	PADINT <i>m,v</i>		VT_ARRAY VT_I4	-2147483648 to 2147483647 for each element*
	PQDINT <i>m,v</i>		VT_ARRAY VT_I4	-2147483648 to 2147483647 for each element*
Real	POREAL <i>m</i>		VT_R4	+/-1.2e-38 to +/-3.4e+38
	PAREAL <i>m</i>		VT_R4	+/-1.2e-38 to +/-3.4e+38
	PQREAL <i>m</i>		VT_R4	+/-1.2e-38 to +/-3.4e+38
Real Array	POREAL <i>m,v</i>		VT_ARRAY VT_R4	+/-1.2e-38 to +/-3.4e+38 for each element*
	PAREAL <i>m,v</i>		VT_ARRAY VT_R4	+/-1.2e-38 to +/-3.4e+38 for each element*
	PQREAL <i>m,v</i>		VT_ARRAY VT_R4	+/-1.2e-38 to +/-3.4e+38 for each element*

\*: For DDE/SuiteLink, the item value is the Hex ASCII representation of the complete array. The result is one string containing all the elements of the array in the Hex ASCII 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 elements (an item in an array), with a range from 1 to (net PDU data size/type size - header information).

\*\*\*: The OI Server provides suffixes (S5T, TR, DT, KT, and BCD) that are used to interpret raw bytes as having a special data format. Review the PLC processor documentation, as not all processors have ladder logic internally to interpret those data formats the same way (for example, the S7-1200 PLC).

---

**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.

---

## Block Items

The SIDirect OI Server supports Block Items for S7-1500 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. See *Conversions and Suffixes of Items (Absolute Addressing)* on page 70 for suffix definitions.

Data Format	Item/Point	Suffix	Data Type	Range
Bit	BLd,Xx.y		VT_BOOL	0 or 1
String	BLd,Sx,v		VT_BSTR	String
	BLd,STRINGx,v		VT_BSTR	String

<b>Data Format</b>	<b>Item/Point</b>	<b>Suffix</b>	<b>Data Type</b>	<b>Range</b>
Byte	BL <i>d</i> ,B <i>x</i>		VT_UI1	0 to 255
	BL <i>d</i> ,BYTE <i>x</i>		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 <i>d</i> ,B <i>x</i> , <i>v</i>		VT_ARRAY VT_UI1	0 to 255 for each element*
	BL <i>d</i> ,BYTE <i>x</i> , <i>v</i>		VT_ARRAY VT_UI1	0 to 255 for each element*
USINT	D<B, > <i>d</i> ,USINT <i>x</i>		VT_UI1	0 to 255
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999***
USINT Array	D<B, > <i>d</i> ,USINT <i>x</i> , <i>v</i>		VT_ARRAY VT_UI1	0 to 255 for each element*
Char	BL <i>d</i> ,CHAR <i>x</i>		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 <i>d</i> ,CHAR <i>x</i> , <i>v</i>		VT_ARRAY VT_I1	-128 to 127 for each element*
SINT	D<B, > <i>d</i> ,SINT <i>x</i>		VT_I1	-128 to 127
		DT	VT_BSTR	1990-1-1-0:00:00.000 to 2089-12-31-23:59:59.999***
SINT Array	D<B, > <i>d</i> SINT <i>x</i> , <i>v</i>		VT_ARRAY VT_UI1	-128 to 127 for each element*
Word	BL <i>d</i> ,W <i>n</i> BL <i>d</i> ,WORD <i>n</i>		VT_UI2	0 to 65535
			VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999
		KT	VT_BSTR	0.0 to 999.3
		S5T	VT_BSTR	0ms to 2h46m30s0ms
		TR	VT_R4	0.0 to 9990.0 (s)
		D	VT_BSTR	1990-1-1 to 2168-12-31
Word Array	BL <i>d</i> ,W <i>n</i> , <i>v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element*
	BL <i>d</i> ,WORD <i>n</i> , <i>v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element*
UINT	D<B, > <i>d</i> ,UINT <i>n</i>		VT_UI2	0 to 65535
		BCD	VT_UI2	0 to 9999***
		KT	VT_BSTR	0.0 to 999.3***
		S5T	VT_BSTR	0ms to 2h46m30s0ms***
		TR	VT_R4	0.0 to 9990.0 (s)***
		D	VT_BSTR	1990-1-1 to 2168-12-31

<b>Data Format</b>	<b>Item/Point</b>	<b>Suffix</b>	<b>Data Type</b>	<b>Range</b>
UINT Array	D<B,l>d,UINT <i>n,v</i>		VT_ARRAY VT_UI2	0 to 65535 for each element*
Integer	BLd,INT <i>n</i>		VT_I2	-32768 to 32767
		BCD	VT_I2	-999 to 999
		D	VT_BSTR	1990-1-1 to 2168-12-31
Integer Array	BLd,INT <i>n,v</i>		VT_BSTR	-32768 to 32767 for each element*
Double Word	BLd,D <i>m</i> BLd,DWORD <i>m</i>		VT_UI4	0 to 4294967295**
			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_648M S to 24D_20H_31M_23S_647M S
Double Word Array	BLd,D <i>m,v</i> BLd,DWORD <i>m,v</i>		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
				0 to 4294967295 for each element**
UDINT	D<B,l>d,UDINT <i>m</i>		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_648M S to 24D_20H_31M_23S_647M S
UDINT Array	D<B,l>d,UDINT <i>m,v</i>		VT_ARRAY VT_UI4	0 to 4294967295 for each element*
Double Integer	BLd,DINT <i>m</i>		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_648M S to 24D_20H_31M_23S_647M S
Double Integer Array	BLd,DINT <i>m,v</i>		VT_ARRAY VT_I4	-2147483648 to 2147483647 for each element
Real	BLd,REAL <i>m</i>		VT_R4	+/-1.2e-38 to +/-3.4e+38

Data Format	Item/Point	Suffix	Data Type	Range
Real Array	BLd,REALm,v		VT_ARRAY VT_R4	+/-1.2e-38 to +/-3.4e+38 for each element*

\*: For DDE/SuiteLink, the item value is the Hex ASCII representation of the complete array. The result is one string containing all the elements of the array in the Hex ASCII 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 elements (an item in an array), 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. See *Conversions and Suffixes of Items (Absolute Addressing)* on page 70 for suffix definitions.

Data Format	Item/Point	Suffix	Data Type	Range
Byte Array	BWd		VT_ARRAY VT_UI1	0 to 255 for each element**,**  Note also that the OI Server does not cache the value of the item written to the PLC.
	BWCd-q,x		VT_ARRAY VT_UI1	0 to 255 for each element**, ***  Note also that the OI Server caches the value of the item written to the PLC.

Data Format	Item/Point	Suffix	Data Type	Range
	BWCd-q.Send		VT_BOOL	TRUE (1) <sup>***, ****</sup> FALSE (0) <sup>***, ****</sup> The value of item (BWCd-q.x), cached by the OI Server, 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 Hex ASCII representation of the complete array. The result is one string containing all the elements of the array in the Hex ASCII 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 OI Server 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.

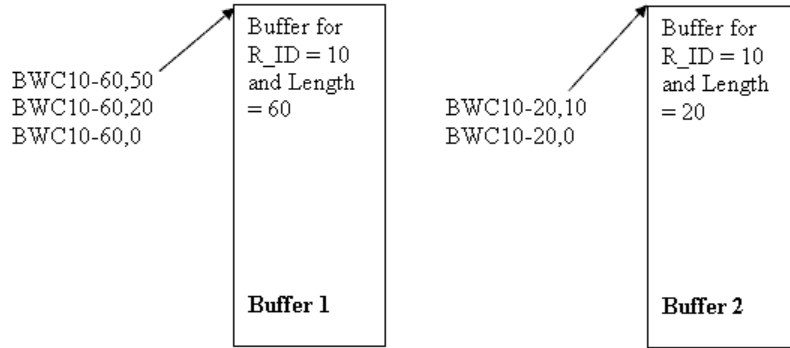
## Block Write Caching

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.

## 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	Cx	None	VT_UI2	0...65535
		Zx	None	VT_UI2
		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.

---

**Note:** S7-1500 PLCs do not support "direct" counter or timer access by the client/server. That is, you cannot use the current S7-300/400 item name Cx or Tx and its associated suffixes to subscribe to counter\timer values from a S7-1500 PLC. However, you can use the database items to indirectly output the associated system counter/timer value (internal IEC counter\timer) from the PLC.

---

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 OI Server is 66.

A 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 OI Server processes a write (**POKE**) to a counter. Although the OI Server allows poking any word value into counters, the S7 PLC can only process values in the range of 0...2457 or 0...999 (BCD).

---

## High-Speed Counters

S7-1500 PLCs support High-Speed Counters. After the High-Speed Counters are enabled in the PLC program, they can be accessed through the Input Bytes in DWord format (for example, ID1000, depending on which counter is to be addressed, the configuration of the counter, and the CPU model of the S7-1500 PLC).

For more information about configuring and using the High-Speed Counters, see the S7-1500 PLC documentation.

## LREAL Data Type and Syntax

S7-1500 PLCs contain a data type named LREAL. The LREAL data type is a ANSI/IEEE 754-1985 64-Bit Floating Point value with a range illustrated in the following table:

Data Type	Bit Size	Range
LREAL	64	-1.7976931348623158e+308 to -2.2250738585072014e-308, ±0, +2.2250738585072014e-308 to +1.7976931348623158e+308

S7-1500 PLCs may not support LREAL types on all memory regions.

The item name syntax for LREAL is as follows:

Memory Type	Syntax
Data Block	D<B,I>d,LREALm D<B,I>d,LREALm,v



<b>Memory Type</b>	<b>Syntax</b>
Flag Bytes	FLREAL <i>m</i>
	MLREAL <i>m</i>
	FLREAL <i>m,v</i>
	MLREAL <i>m,v</i>
Input Bytes	ILREAL <i>m</i>
	ELREAL <i>m</i>
	ILREAL <i>m,v</i>
	ELREAL <i>m,v</i>
Output Bytes	OLREAL <i>m</i>
	ALREAL <i>m</i>
	QLREAL <i>m</i>
	OLREAL <i>m,v</i>
	ALREAL <i>m,v</i>
	QLREAL <i>m,v</i>
Peripheral Input Bytes	PILREAL <i>m</i>
	PELREAL <i>m</i>
	PILREAL <i>m,v</i>
	PELREAL <i>m,v</i>
Peripheral Output Bytes	POLREAL <i>m</i>
	PALREAL <i>m</i>
	PQLREAL <i>m</i>
	POLREAL <i>m,v</i>
	PALREAL <i>m,v</i>
	PQLREAL <i>m,v</i>

Where:

- d is the data block number, with a range from 1 to 65535
- m is the start offset of an LREAL tag

v is the length of an array in elements, with a range from 1 to 65535

## DTL Data Type and Syntax

The S7-1500 PLCs contain a data type named DTL (Date Time Long). The DTL can be subscribed as a string (VT\_BSTR), which provides the full range and resolution of 1 nanosecond, or as a date and time (VT\_DATE), which provides the full range but with the highest resolution of 1 second. The DTL data type in the PLC is a 12-byte structure with the following format:

Byte	Field Name	Date Type	Value Range
0,1	Year	UINT	1970 to 2554
2	Month	USINT	1 to 12
3	Day	USINT	1 to 31
4	Weekday	USINT	1 (Sunday) to 7 (Saturday)
5	Hour	USINT	0 to 23
6	Minute	USINT	0 to 59
7	Seconds	USINT	0 to 59
8,9,10,11	Nanoseconds	UDINT	0 to 999,999,999

The OI Server reads the 12-byte structure from the S7 PLC and converts it into one of the following canonical types based on the item name suffix specified:

DTL Suffix	Canonical Type	Range
<none>	VT_BSTR	1970-1-1-0:0:0.000000000 to 2554-12-31-23:59:59.999999999 (Highest resolution in units of 1 nanoseconds increments)
TREAL	VT_DATE	1970-1-1-0:0:0 to 2554-12-31-23:59:59 (Highest resolution is in units of 1 sec increments) The syntax for VT_DATE can be YYYY/MM/DD, MM/DD/YYYY, or YYYY-MM-DD.

The item name syntax for DTL is as follows:

<b>Memory Type</b>	<b>Syntax</b>	<b>Canonical Type</b>
Data Block	D<B,l>d,DTLm	VT_BSTR
	D<B,l>d,DTLmTREAL	VT_DATE
	D<B,l>d,DTLm,v	VT_ARRAY  VT_DATE
	D<B,l>d,DTLm,vTREAL	VT_ARRAY  VT_DATE
Flag Bytes	FDTLm	VT_BSTR
	MDTLm	VT_BSTR
	FDTLmTREAL	VT_DATE
	MDTLmTREAL	VT_DATE
	FDTLm,v	VT_ARRAY  VT_BSTR
	MDTLm,v	VT_ARRAY  VT_BSTR
	FDTLm,vTREAL	VT_ARRAY  VT_DATE
	MDTLm,vTREAL	VT_ARRAY  VT_DATE
Input Bytes	IDTLm	VT_BSTR
	EDTLm	VT_BSTR
	IDTLmTREAL	VT_DATE
	EDTLmTREAL	VT_DATE
	IDTLm,v	VT_ARRAY  VT_BSTR
	EDTLm,v	VT_ARRAY  VT_BSTR
	IDTLm,vTREAL	VT_ARRAY  VT_DATE
	EDTLm,vTREAL	VT_ARRAY  VT_DATE

<b>Memory Type</b>	<b>Syntax</b>	<b>Canonical Type</b>	
Output Bytes	ODTL <i>m</i>	VT_BSTR	
	ADTL <i>m</i>	VT_BSTR	
	QDTL <i>m</i>	VT_BSTR	
	ODTL <i>m</i> TREAL	VT_DATE	
	ADTL <i>m</i> TREAL	VT_DATE	
	QDTL <i>m</i> TREAL	VT_DATE	
	ODTL <i>m,v</i>	VT_ARRAY  VT_BSTR	
	ADTL <i>m,v</i>	VT_ARRAY  VT_BSTR	
	QDTL <i>m,v</i>	VT_ARRAY  VT_BSTR	
	ODTL <i>m,v</i> TREAL	VT_ARRAY  VT_DATE	
	ADTL <i>m,v</i> TREAL	VT_ARRAY  VT_DATE	
	QDTL <i>m,v</i> TREAL	VT_ARRAY  VT_DATE	
	Peripheral Input Bytes	PIDTL <i>m</i>	VT_BSTR
		PEDTL <i>m</i>	VT_BSTR
PIDTL <i>m</i> TREAL		VT_DATE	
PEDTL <i>m</i> TREAL		VT_DATE	
PIDTL <i>m,v</i>		VT_ARRAY  VT_BSTR	
PEDTL <i>m,v</i>		VT_ARRAY  VT_BSTR	
PIDTL <i>m,v</i> TREAL		VT_ARRAY  VT_DATE	
PEDTL <i>m,v</i> TREAL		VT_ARRAY  VT_DATE	

<b>Memory Type</b>	<b>Syntax</b>	<b>Canonical Type</b>
Peripheral Output Bytes	PODTL <i>m</i>	VT_BSTR
	PADTL <i>m</i>	VT_BSTR
	PQDTL <i>m</i>	VT_BSTR
	PODTL <i>m</i> TREAL	VT_DATE
	PADTL <i>m</i> TREAL	VT_DATE
	PQDTL <i>m</i> TREAL	VT_DATE
	PODTL <i>m,v</i>	VT_ARRAY  VT_BSTR
	PADTL <i>m,v</i>	VT_ARRAY  VT_BSTR
	PQDTL <i>m,v</i>	VT_ARRAY  VT_BSTR
	PODTL <i>m,v</i> TREAL	VT_ARRAY  VT_DATE
	PADTL <i>m,v</i> TREAL	VT_ARRAY  VT_DATE
	PQDTL <i>m,v</i> TREAL	VT_ARRAY  VT_DATE

Where:

- d is the data block number, with a range from 1 to 65535
- m is the start offset of an DTL tag
- v is the length of an array in elements (an item in an array), with a range from 1 to 65535

### DTL Clamping

Any DTL value being read from or poked to the PLC that has a year field less than the minimum or greater than the maximum year results in the date and time being clamped as follows:

For Canonical VT\_BSTR:

Year < 1970 = 1970-01-01-00:00:00.000000000

Year > 2554 = 2554-12-31-23:59:59.999999999

For Canonical VT\_DATE:

Year < 1970 = 1970-01-01-00:00:00

Year > 2554 = 2554-12-31-23:59:59

If any other field, month, day, hour, min, second, or nanosecond is less than the minimum or greater than the maximum, it results in a rejected poke or read. Also, an uncertain quality results for subscribed tags.

This applies to the syntax of all DTL names.

## Using DTL without a Suffix

When poking or subscribing a DTL without a suffix where the RequestedType=VT\_BSTR and the CanonicalType=VT\_BSTR and you do not enter the entire 9 digits of the nanosecond field, the field is padded to the right with zeros.

The reason for doing this is to maintain the meaning of the decimal place after the seconds field. For example, 2011-10-31-1:1:1.5 results in 500000000 being poked to the nanoseconds field of the DTL, which really means 1/2 a second. To enter 5 nanoseconds, you must poke 2011-10-31-1:1:1.000000005.

All other fields of the date and time are considered to be padded to the left.

## Using DTL with a TREAL Suffix

When poking or subscribing a DTL using the TREAL suffix where the RequestedType=VT\_BSTR and the CanonicalType=VT\_DATE, the string passed between the SIdirect OI Server and the client is formatted using the "short date" and "long time" format that you configure in the region and language settings of the computer.

## Conversions and Suffixes of Items (Absolute Addressing)

This section describes what data-format items and suffixes are converted and what they are converted into. The items and suffixes described here apply to the SIdirect Legacy object (absolute addressing). For items and suffixes that apply to the SIdirect Symbolic object, see *Conversions and Suffixes of Items (Symbolic Addressing)* on page 84.

### 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 the meaning is the same. This means a memory content of 0x3999 (as in the example for KT) results in the string of 2h46m30s0ms.

The meaning of 999.3 (KT) is:

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 the meaning is the same. 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 OI Server 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.

---

## Generic OPC Syntax

An OI Server 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 devices 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 OI Server-dependent. In OPC data acquisition OI Servers, 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 OI Servers, Access Paths are equivalent to Device Groups. This parameter defines the update interval between the OI Server and the field device for accessing the values of data points in the PLC.

## VT\_Array Syntax in Application Server

VT\_Array items from the SIDirect OI Server can fail to return good VTQ data in Object Viewer without the addition of an index to the Item/Point syntax. While the syntax provided in this section is correct, VT\_Arrays require a [-1] offset. Without this offset, the VT\_Array will not count the 0 byte in the range.

Following is an example of the syntax for a VT\_Array item to receive proper validation under a LMX client such as Object Viewer in Application Server:

```
S7C_S7_001.SG1.attribute(DB111,WORD10,4)[-1]
```

In this example, the syntax has the following structure:

```
<SIDirect_DiDevice_Instance>.<ScanGroup>.attribute(D<B>d,WORDn,v)[-1]
```

where

- d is the data block number,
- n is the start address of a 2-byte data array, and
- v is the length of data in elements (the size of the array). Each item in an array is called an element.

## SIDirect Symbolic Object Reference (Symbolic Addressing)

This section provides reference information specific to the SIDirect Symbolic object. Use the Symbolic object when symbolic addressing is used to communicate with the S7-1500 PLC.

### SIDirect Symbolic Naming Convention

The Symbolic interface included with the SIDirect OI Server supports symbolic addressing, whether or not Optimized Block Access has been enabled in the Siemens S7-1500 PLC. The tag names used in this hierarchy correspond to the symbolic names that you configure in the PLC.

---

**Note:** Symbolic addressing is NOT supported through the SIDirect Legacy interface.

---

The naming conventions used in the following sections about data types are:

- |                     |                               |
|---------------------|-------------------------------|
| [ square brackets ] | Alphanumeric name is required |
| < angle brackets >  | Alphanumeric name is optional |

{ curly brackets }                      Naming fields within the brackets can be recursive

## Basic Data Types

Basic data types for symbolic addressing include the following items:

- Bit data block items
- Analog data block items
- String data block items

```
[DataBlockName].[SymbolicTagName]
```

Example:

```
DB5_INT.TAG_INT
```

```
BatchProcess_Block.EmergencyStartStop
```

## Complex Data Types

Complex data types for symbolic addressing include the following items:

- Array type items
- Structure data block items

```
[DataBlockName].[SymbolicTagName]<[ArrayIndex]>.<MemberName>
```

Example:

```
DB9_STRING.TAG_STRING
```

```
Kettle2.PigmentTube[4]
```

```
Data_block_Items.IEC_Timer1.CV
```

## User Defined Data Types

User defined data types for symbolic addressing include the following item:

- User Defined Type (UDT) items

```
[DataBlockName].[StructTagName]<[ArrayIndex]>.[MemberName]<[ArrayIndex]>
```

Example:

```
ProductionRoom.Reactor1.Concentrate_Pump.Motor_OL
```

```
FoodPlant.CookDeck[1].IngredientTanks[2,5].InletValve
```

## System Data Types

You should observe the following item syntax constraints when using symbolic addressing:

- Input, Output, Memory area items
- Counter and Timer items

```
[PLCTagName]
```

Example:

PLCTag_Byte	PLC Tag with reference to Memory Area item “%MB1”
PLCTag_Timer	PLC Tag with reference to Timer Area item “%T0”

## General SIDirect Item Syntax Constraints

You should observe the following item syntax constraints when using symbolic addressing:

- Maximum length of a symbolic name: 128 characters in the Data Block Table
- Maximum dimension of an array: 6
- Maximum number of elements per array: 1600
- Maximum nesting depth of a structure item: 8 levels
- Maximum of number of components in a structure: 252
- Symbolic names can use any character, with the following exceptions:
  - Space character ( )
  - Period character (.)
  - Double quote character (")

## Data Types in the TIA Portal

The TIA Portal for S7-1500 PLCs defines a large number of data types and data formats. These pertain to how the S7-1500 PLC handles data transfer internally. Data types, data formats, structural elements, suffixes, and value ranges for items using symbolic addressing in S7-1500 PLCs are described in the following sections.

Data types can be divided into three categories:

- *General Data Types* on page 75: all basic data types defined in the controller for general purpose usage.
- *Counter/Timer Data Types* on page 78: all structures and parameters provided by the system for counters and timers.
- *Special Data Types* on page 82: used for special purposes through specific instructions built into the controller.

## General Data Types

General data types include all basic data types defined in the controller for general purpose usage. See *Conversions and Suffixes of Items (Symbolic Addressing)* on page 84 for suffix definitions.

---

**Note:** Structural element names are predefined by the system and cannot be changed.

---

Data Format	Structural Element	Suffix	Data Type	Access	Value Range / Notes
Bool			VT_BOOL	Read/Write	FALSE or TRUE
Byte			VT_UI1	Read/Write	0 to 255

Char			VT_UI1	Read/Write	ASCII character set
DInt			VT_I4	Read/Write	-2147483648 to +2147483647
DWord			VT_UI4	Read/Write	0 to 4294967295
Int			VT_I2	Read/Write	-32768 to 32767
LInt			VT_I8	Read/Write	-9223372036854775808 to +9223372036854775807
LReal			VT_R8	Read/Write	-1.7976931348623158e+308 to -2.2250738585072014e-308 ±0,0 +2.2250738585072014e-308 to +1.7976931348623158e+308
LWord			VT_UI8	Read/Write	0 to 18446744073709551615
Real			VT_R4	Read/Write	-3.402823e+38 to -1.175495e-38 ±0,0 +1.175495e-38 to +3.402823e+38
SInt			VT_I1	Read/Write	-128 to 127
String			VT_BSTR	Read/Write	0 to 254 characters
UDInt			VT_UI4	Read/Write	0 to 4294967295
UInt			VT_UI2	Read/Write	0 to 65535
ULInt			VT_UI8	Read/Write	0 to 18446744073709551615
USInt			VT_UI1	Read/Write	0 to 255
WChar			VT_UI2	Read/Write	\$0000 - \$D7FF
WString			VT_BSTR	Read/Write	0 to 254 characters x. possible values: 0 to 16382
Word			VT_UI2	Read/Write	0 to 65535
Date			VT_UI2	Read/Write	0 to 65535 (days)
		D	VT_BSTR	Read only	1990-01-01 to 2168-12-31
Date_And_Time			VT_Date	Read/Write	1990-01-01-00:00:00.000 to 2089-12-31-23:59:59.999
		DT	VT_BSTR	Read/Write	1990-01-01-00:00:00.000 to 2089-12-31-23:59:59.999 <b>Note:</b> DT suffix is required for SuiteLink.

LDT			VT_UI8	Read/Write	0 to 18446744073709551615 (nanoseconds)
		LDT	VT_BSTR	Read only	1970-01-01-0:0:0.000000000 to 2263-04-11-23:47:16.854775808
LTime			VT_I8	Read/Write	-9223372036854775808 to +9223372036854775807 (nanoseconds)
		LT	VT_BSTR	Read only	-106751d23h47m16s854ms775us808ns to 106751d23h47m16s854ms775us807ns
LTime_Of_Day			VT_UI8	Read/Write	0 to 18446744073709551615
S5Time			VT_I4	Read/Write	0 to 9990000 (milliseconds)
		S5T	VT_BSTR	Read only	10MS to 9S_990MS (Resolution: 0.01s)
					100MS to 1MIN_39S_900MS (Resolution: 0.1s)
					1S to 16MIN_39S (Resolution: 1s)
	0MS to 2H_46M_30S_0MS (Resolution: 10s <Default>)				
Time			VT_I4	Read/Write	-2147483648 to +2147483647 (milliseconds)
		T	VT_BSTR	Read only	-24d20h31m23s648ms to +24d20h31m23s647ms
Time_Of_Day			VT_UI4	Read/Write	0 to 4294967295 (milliseconds)
		TOD	VT_BSTR	Read only	00:00:00.000 to 23:59:59.999
DTL <Structure>					
UI Int	.YEAR		VT_UI2	Read only	1970 to 2262
US Int	.MONTH		VT_UI1	Read only	1 to 12
US Int	.DAY		VT_UI1	Read only	1 to 31
US Int	.WEEKDAY		VT_UI1	Read only	1(Sunday) to 7(Saturday)

US Int	.HOUR		VT_UI1	Read only	0 to 23
US Int	.MINUTE		VT_UI1	Read only	0 to 59
US Int	.SECOND		VT_UI1	Read only	0 to 59
UD Int	.NANOSECO ND		VT_UI4	Read only	0 to 999999999

## Counter/Timer Data Types

Counter/Timer includes all structures and parameters for counters and timers provided by the system.

---

**Note:** Structural element names are predefined by the system and cannot be changed.

---

Data Format	Structural Element	Suffix	Data Type	Access	Value Range / Notes
Counter			VT_UI2	Read/Write	0 to 999
Timer			VT_I4	Read/Write	0 to 999000
IEC_COUNTER			<Structure>	N/A	
Bool	.CU		VT_BOOL	Read/Write	FALSE or TRUE (Count Up)
Bool	.CD		VT_BOOL	Read/Write	FALSE or TRUE (Count Down)
Bool	.R		VT_BOOL	Read/Write	FALSE or TRUE (Reset)
Bool	.LD		VT_BOOL	Read/Write	FALSE or TRUE (Load)
Bool	.QU		VT_BOOL	Read/Write	FALSE or TRUE (Status of Up Count)
Bool	.QD		VT_BOOL	Read/Write	FALSE or TRUE (Status of Down Count)
Int	.PV		VT_I2	Read/Write	-32768 to 32767 (Preset Counter Value)
Int	.CV		VT_I2	Read/Write	-32768 to 32767 (Current Counter Value)
IEC_DCOUNTER			<Structure>	N/A	
Bool	.CU		VT_BOOL	Read/Write	FALSE or TRUE (Count Up)
Bool	.CD		VT_BOOL	Read/Write	FALSE or TRUE (Count Down)
Bool	.R		VT_BOOL	Read/Write	FALSE or TRUE (Reset)

Bool	.LD		VT_BOOL	Read/Write	FALSE or TRUE (Load)
Bool	.QU		VT_BOOL	Read/Write	FALSE or TRUE (Status of Up Count)
Bool	.QD		VT_BOOL	Read/Write	FALSE or TRUE (Status of Down Count)
DInt	.PV		VT_I4	Read/Write	-2147483648 to +2147483647 (Preset Counter Value)
DInt	.CV		VT_I4	Read/Write	-2147483648 to +2147483647 (Current Counter Value)
IEC_LCOUNTER			<Structure>	N/A	
Bool	.CU		VT_BOOL	Read/Write	FALSE or TRUE (Count Up)
Bool	.CD		VT_BOOL	Read/Write	FALSE or TRUE (Count Down)
Bool	.R		VT_BOOL	Read/Write	FALSE or TRUE (Reset)
Bool	.LD		VT_BOOL	Read/Write	FALSE or TRUE (Load)
Bool	.QU		VT_BOOL	Read/Write	FALSE or TRUE (Status of Up Count)
Bool	.QD		VT_BOOL	Read/Write	FALSE or TRUE (Status of Down Count)
LInt	.PV		VT_I8	Read/Write	-9223372036854775808 to +9223372036854775807 (Preset Counter Value)
LInt	.CV		VT_I8	Read/Write	-9223372036854775808 to +9223372036854775807 (Current Counter Value)
IEC_SCOUNTER			<Structure>	N/A	
Bool	.CU		VT_BOOL	Read/Write	FALSE or TRUE (Count Up)
Bool	.CD		VT_BOOL	Read/Write	FALSE or TRUE (Count Down)
Bool	.R		VT_BOOL	Read/Write	FALSE or TRUE (Reset)
Bool	.LD		VT_BOOL	Read/Write	FALSE or TRUE (Load)
Bool	.QU		VT_BOOL	Read/Write	FALSE or TRUE (Status of Up Count)
Bool	.QD		VT_BOOL	Read/Write	FALSE or TRUE (Status of Down Count)
SInt	.PV		VT_I1	Read/Write	-128 to 127 (Preset Counter Value)

SInt	.CV		VT_I1	Read/Write	-128 to 127 (Current Counter Value)
IEC_UCOUNTER			<Structure>	N/A	
Bool	.CU		VT_BOOL	Read/Write	FALSE or TRUE (Count Up)
Bool	.CD		VT_BOOL	Read/Write	FALSE or TRUE (Count Down)
Bool	.R		VT_BOOL	Read/Write	FALSE or TRUE (Reset)
Bool	.LD		VT_BOOL	Read/Write	FALSE or TRUE (Load)
Bool	.QU		VT_BOOL	Read/Write	FALSE or TRUE (Status of Up Count)
Bool	.QD		VT_BOOL	Read/Write	FALSE or TRUE (Status of Down Count)
UInt	.PV		VT_UI2	Read/Write	0 to 65535 (Preset Counter Value)
UInt	.CV		VT_UI2	Read/Write	0 to 65535 (Current Counter Value)
IEC_UDCOUNTER			<Structure>	N/A	
Bool	.CU		VT_BOOL	Read/Write	FALSE or TRUE (Count Up)
Bool	.CD		VT_BOOL	Read/Write	FALSE or TRUE (Count Down)
Bool	.R		VT_BOOL	Read/Write	FALSE or TRUE (Reset)
Bool	.LD		VT_BOOL	Read/Write	FALSE or TRUE (Load)
Bool	.QU		VT_BOOL	Read/Write	FALSE or TRUE (Status of Up Count)
Bool	.QD		VT_BOOL	Read/Write	FALSE or TRUE (Status of Down Count)
UDInt	.PV		VT_UI4	Read/Write	0 to 4294967295 (Preset Counter Value)
UDInt	.CV		VT_UI4	Read/Write	0 to 4294967295 (Current Counter Value)
IEC_ULCOUNTER			<Structure>	N/A	
Bool	.CU		VT_BOOL	Read/Write	FALSE or TRUE (Count Up)
Bool	.CD		VT_BOOL	Read/Write	FALSE or TRUE (Count Down)
Bool	.R		VT_BOOL	Read/Write	FALSE or TRUE (Reset)



Bool	.LD		VT_BOOL	Read/Write	FALSE or TRUE (Load)
Bool	.QU		VT_BOOL	Read/Write	FALSE or TRUE (Status of Up Count)
Bool	.QD		VT_BOOL	Read/Write	FALSE or TRUE (Status of Down Count)
ULInt	.PV		VT_UI8	Read/Write	0 to 18446744073709551615 (Preset Counter Value)
ULInt	.CV		VT_UI8	Read/Write	0 to 18446744073709551615 (Current Counter Value)
IEC_USCOUNTER			<Structure>	N/A	
Bool	.CU		VT_BOOL	Read/Write	FALSE or TRUE (Count Up)
Bool	.CD		VT_BOOL	Read/Write	FALSE or TRUE (Count Down)
Bool	.R		VT_BOOL	Read/Write	FALSE or TRUE (Reset)
Bool	.LD		VT_BOOL	Read/Write	FALSE or TRUE (Load)
Bool	.QU		VT_BOOL	Read/Write	FALSE or TRUE (Status of Up Count)
Bool	.QD		VT_BOOL	Read/Write	FALSE or TRUE (Status of Down Count)
USInt	.PV		VT_UI1	Read/Write	0 to 255 (Preset Counter Value)
USInt	.CV		VT_UI1	Read/Write	0 to 255 (Current Counter Value)
IEC_TIMER			<Structure>	N/A	
Time	.ST		VT_I4	Read/Write	-2147483648 to +2147483647 (Start Time)
Time	.PT		VT_I4	Read/Write	-2147483648 to +2147483647 (Present Time)
Time	.ET		VT_I4	Read/Write	-2147483648 to +2147483647 (Elapsed Time)
Bool	.RU		VT_BOOL	Read/Write	FALSE or TRUE (Running Status)
Bool	.IN		VT_BOOL	Read/Write	FALSE or TRUE (Start Input)
Bool	.Q		VT_BOOL	Read/Write	FALSE or TRUE (Pulse Output)
IEC_LTIMER			<Structure>	N/A	
LTime	.ST		VT_I8	Read/Write	-9223372036854775808 to +9223372036854775807 (Start Time)

LTime	.PT		VT_I8	Read/Write	-9223372036854775808 to +9223372036854775807 (Present Time)
LTime	.ET		VT_I8	Read/Write	-9223372036854775808 to +9223372036854775807 (Elapsed Time)
Bool	.RU		VT_BOOL	Read/Write	FALSE or TRUE (Running Status)
Bool	.IN		VT_BOOL	Read/Write	FALSE or TRUE (Start Input)
Bool	.Q		VT_BOOL	Read/Write	FALSE or TRUE (Pulse Output)

## Special Data Types

Special data types include:

- System data type
- Hardware data type

These are used for special purposes through specific instructions in the controller. Their availability depends on the model of the controller. Some frequently used special data types are listed in the following table.

---

**Note:** Structure element names are predefined by the system and cannot be changed.

---

Data Format	Structural Element	Suffix	Data Type	Access	Value Range / Notes
AOM_IDENT			VT_UI4	Read/Write	0 to 4294967295
CONN_ANY			VT_UI2	Read/Write	0 to 65535
CONN_OUC			VT_UI2	Read/Write	0 to 65535
CONN_PRG			VT_UI2	Read/Write	0 to 65535
CONN_R_ID			VT_UI4	Read/Write	0 to 4294967295
CREF	<Structure>		N/A		
Byte	.BLOCK_TYPE		VT_UI1	Read/Write	0 to 255
UInt	.CB_NUMBER		VT_UI2	Read/Write	0 to 65535
UDInt	.OFFSET		VT_UI4	Read/Write	0 to 4294967295
DB_ANY			VT_UI2	Read/Write	0 to 65535
DB_WWW			VT_UI2	Read/Write	0 to 65535
EVENT_ANY			VT_UI4	Read/Write	0 to 4294967295
EVENT_ATT			VT_UI4	Read/Write	0 to 4294967295
EVENT_HWINT			VT_UI4	Read/Write	0 to 4294967295

ErrorStruct	<Structure>		N/A		
Word	.ERROR_ID		VT_UI1	Read/Write	0 to 255
Byte	.FLAGS		VT_UI1	Read/Write	0 to 255
Byte	.REACTION		VT_UI1	Read/Write	0 to 255
Byte	.MODE		VT_UI2	Read/Write	0 to 255
UInt	.OPERAND_NUMBER		VT_UI2	Read/Write	1970 to 2262
UInt	.POINTER_NUMBER_LOCATION		VT_UI1	Read/Write	1970 to 2262
UInt	.SLOT_NUMBER_SCOPE		VT_UI2	Read/Write	1970 to 2262
CREF	<Structure>		N/A		
Byte	.CREF.BLOCK_TYPE		VT_UI4	Read/Write	0 to 4294967295
UInt	.CREF.CB_NUMBER		VT_UI1	Read/Write	1970 to 2262
UDInt	.CREF.OFFSET		VT_UI2	Read/Write	0 to 999999999
NREF	<Structure>		N/A		
Byte	.NREF.AREA		VT_UI1	Read/Write	0 to 255
UInt	.NREF.DB_NUMBER		VT_UI2	Read/Write	1970 to 2262
UDInt	.NREF.OFFSET		VT_UI4	Read/Write	0 to 4294967295
HW_ANY			VT_UI2	Read/Write	0 to 65535
HW_DEVICE			VT_UI2	Read/Write	0 to 65535
HW_DPMASTER			VT_UI2	Read/Write	0 to 65535
HW_DPSLAVE			VT_UI2	Read/Write	0 to 65535
HW_HSC			VT_UI2	Read/Write	0 to 65535
HW_IEPORT			VT_UI2	Read/Write	0 to 65535
HW_INTERFACE			VT_UI2	Read/Write	0 to 65535
HW_IO			VT_UI2	Read/Write	0 to 65535
HW_IOSYSTEM			VT_UI2	Read/Write	0 to 65535
HW_MODULE			VT_UI2	Read/Write	0 to 65535
HW_PTO			VT_UI2	Read/Write	0 to 65535
HW_PWM			VT_UI2	Read/Write	0 to 65535
HW_SUBMODULE			VT_UI2	Read/Write	0 to 65535
NREF	<Structure>		N/A		
Byte	.NREF.AREA		VT_UI1	Read/Write	0 to 255

UInt	.NREF.DB_NUMBER		VT_UI2	Read/Write	0 to 65535
UDInt	.NREF.OFFSET		VT_UI4	Read/Write	0 to 4294967295
OB_ANY			VT_I2	Read/Write	-32768 to 32767
OB_ATT			VT_I2	Read/Write	-32768 to 32767
OB_CYCLIC			VT_I2	Read/Write	-32768 to 32767
OB_DELAY			VT_I2	Read/Write	-32768 to 32767
OB_DIAG			VT_I2	Read/Write	-32768 to 32767
OB_HWINT			VT_I2	Read/Write	-32768 to 32767
OB_PCYCLE			VT_I2	Read/Write	-32768 to 32767
OB_STARTUP			VT_I2	Read/Write	-32768 to 32767
OB_TIMEERROR			VT_I2	Read/Write	-32768 to 32767
OB_TOD			VT_I2	Read/Write	-32768 to 32767
PIP			VT_UI2	Read/Write	0 to 65535
PORT			VT_UI2	Read/Write	0 to 65535
RTM			VT_UI2	Read/Write	0 to 65535
STRUCT	<Structure>		N/A		
The STRUCT data type does not have a pre-defined name or member names. It represents a data structure composed of different data types. Any data type can be included in STRUCT. See the Siemens programming guidelines for additional information					

## Conversions and Suffixes of Items (Symbolic Addressing)

- This section describes what data-format items and suffixes are converted and what they are converted into. The items and suffixes described here apply to the SIDirect Symbolic object (symbolic addressing). For items and suffixes that apply to the SIDirect Legacy object, see *Conversions and Suffixes of Items (Absolute Addressing)* on page 70

### Suffix D (Symbolic)

The suffix D can be used with the DATE data type. When used without the suffix D, DATE saves the date as an unsigned integer, containing the year, month, and day. See *General Data Types* on page 75 for additional information about DATE.

The suffix D (Date) converts DATE into a string to store the year, month, and day.

Usage: **DATE D**

For example: MyDate D

- The suffix D converts DATE to an 8-byte string (VT\_BSTR) that stores the date. Range is 1990-01-01 to 2168-12-31. The construction is the number of days since 1/1/1990.
- When used without the suffix D, the SIDirect OI Server reads DATE from the PLC as an unsigned integer (VT\_UI2). Range is 0 to 65535 (days).

## Suffix LDT (Symbolic)

The suffix LDT can be used with the LDT data type. The LDT data type, when used without the suffix LDT, saves the date as an unsigned integer. See *General Data Types* on page 75 for additional information about LDT.

The suffix LDT converts the LDT data type into a string.

Usage: **LDT LDT**

For example: `MyLDT LDT`

- The suffix LDT converts the LDT data type into a 12-byte string (V\_BSTR) that stores the date and time in string format. Range is 1970-01-01-0:0:0.000000000 to 2263-04-11-23:47:16.854775808. The construction is the number of nanoseconds since 1/1/1970 0:0.
- When used without the suffix LDT, the SIDirect OI Server reads data type LDT from the PLC as an unsigned integer (VT\_UI8). Range is 0 to 18446744073709551615 (nanoseconds)

## Suffix LT (Symbolic)

The suffix LT can be used with the LTime data type. When used without the suffix, LTime saves time as an integer, containing number of days (d), hours (h), minutes (m), seconds (s), milliseconds (ms), microseconds, (us), and nanoseconds (ns). See *General Data Types* on page 75 for additional information about LTime.

The suffix LT converts the LTime data type into a 64-bit string defining the time.

Usage: **LTime LT**

For example: `MyLTime LT`

- The suffix LT converts LTime into a 64-bit string (VT\_BSTR) that stores the time in string format. Range is -106751d23h47m16s854ms775us808ns to 106751d23h47m16s854ms775us807ns
- When used without the suffix LT, the SIDirect OI Server reads LTime from the PLC as an integer (VT\_I8). Range is -9223372036854775808 to +9223372036854775807

## Suffix S5T (Symbolic)

The suffix S5T can be used with the S5Time data type. When used without the suffix S5T, the SIDirect OI Server reads the 2-byte time from the S7-1500 PLC. See *General Data Types* on page 75 for additional information about S5Time.

The suffix S5T converts S5Time into a string.

Usage: **S5Time S5T**

For example: `MyS5Time S5T`

- The suffix S5T converts S5Time into string (VT\_BSTR) that stores the time as a value in the range of 0ms to 2h46m30s.
- When used without the suffix D, the SIDirect OI Server reads S5Time from the PLC as an unsigned integer (VT\_UI2) with a range of 0 to 65535 (milliseconds).

## Suffix T (Symbolic)

The suffix T can be used with the Time data type. When used without the suffix S5T, the SIDirect OI Server reads the 4-byte time from the S7-1500 PLC. See *General Data Types* on page 75 for additional information about Time.

The suffix T converts Time into a string.

**Usage: Time T**

For example: `MyTime T`

- The suffix T converts Time into a string (VT\_BSTR) that stores the time as a value in the range of -24d20h31m23s648ms to +24d20h31m23s647ms.
- When used without the suffix T, the SIdirect OI Server reads Time from the PLC as an integer (VT\_I4) with a range of -2147483648 to +2147483647 (milliseconds).

**Suffix TOD (Symbolic)**

The suffix TOD can be used with the Time\_Of\_Day data type. When used without the suffix TOD, the SIdirect OI Server reads the 4-byte time from the S7-1500 PLC. See *General Data Types* on page 75 for additional information about Time\_Of\_Day.

The suffix TOD converts Time\_Of\_Day to a string.

**Usage: Time\_Of\_Day TOD**

For example: `MyTime_Of_Day TOD`

- The suffix TOD converts Time\_Of\_Day to a string (VT\_BSTR) that stores the time as a value in the range of 0:00:00.000 to 23:59:59.999.
- When used without the suffix TOD, the SIdirect OI Server reads Time\_Of\_Day from the PLC as an unsigned 4-byte integer (VT\_UI2) with a range of 0 to 4294967295 (milliseconds).

**SIdirect Driver Diagnostic Info Items**

The SIdirect OI Server provides Diagnostic System items specific to the Symbolic object. Diagnostic Info Items are the group of items that provide statistics information to reflect the performance of the driver.

Diagnostic Info Item names start with "\$SYS\$" to differentiate them from PLC tags.

Item Name	Data Type	Item Description	Access	Range/ Format
\$SYS\$ItemUpdateRate	VT_UI4	Returns the instantaneous updates per second for subscribed items.  To reset the value, set system item \$SYS\$ResetStatistics to 1.	Read only	Range: 0 to 4294967295

\$SYS\$Load	VT_R4	<p>Returns the load of the connection to the PLC. The load is the rate of item updates over the last 15 minutes, and is expressed as a multiplier of the Load Factor.</p> <p>To reset the value, set system item \$SYS\$ResetStatistics to 1.</p> <p>A load factor of 1 indicates that the rate of item updates is 1000 values/second; a load factor of 2 indicates that the rate of item updates is 2000 values/second.</p>	Read only	
\$SYS\$TagsResolved	VT_UI4	<p>Returns the number of unique tags that have been sent successfully to the PLC for update.</p>	Read only	Range: 0 to 4294967295
\$SYS\$ReadCount	VT_UI4	<p>Returns the cumulative number of values that have been received successfully from the PLC.</p> <p>To reset the value, set system item \$SYS\$ResetStatistics to 1.</p>	Read only	Range: 0 to 4294967295
\$SYS\$WriteCount	VT_UI4	<p>Returns the cumulative number of values that have been written successfully to the PLC.</p> <p>To reset the value, set system item \$SYS\$ResetStatistics to 1.</p>	Read only	Range: 0 to 4294967295
\$SYS\$ReadCountFailed	VT_UI4	<p>Returns the cumulative number of failed read requests sent from the server to the PLC.</p> <p>To reset the value, set system item \$SYS\$ResetStatistics to 1.</p>	Read only	Range: 0 to 4294967295

\$SYS\$WriteCountFailed	VT_UI4	Returns the cumulative number of failed write requests sent from the server to the PLC.  To reset the value, set system item \$SYS\$ResetStatistics to 1.	Read only	Range: 0 to 4294967295
\$SYS\$LastReadDuration	VT_UI4	Returns the elapsed time in milliseconds for the most recent read request to the PLC.  Quality is set to 0x08 if a read has not occurred since the server started up.	Read only	Range: 0 to 4294967295
\$SYS\$LastWriteDuration	VT_UI4	Returns the elapsed time in milliseconds for the most recent write request to the PLC.  Quality is set to 0x08 if a write has not occurred since the server started up.	Read only	Range: 0 to 4294967295
\$SYS\$LastReadCount	VT_UI4	Returns the total number of tags that were involved in the most recent read request to the PLC.	Read only	Range: 0 to 4294967295
\$SYS\$LastWriteCount	VT_UI4	Returns the total number of tags that were involved in the most recent write request to the PLC.	Read only	Range: 0 to 4294967295
\$SYS\$HostName	VT_BSTR	Returns the configured host name or IP address of PLC.	Read only	
\$SYS\$LastReadTime	VT_DATE	Returns the system time of the most recent read request to the PLC.  Quality is set to 0x08 if a read has not occurred since the server started up.	Read only	System time is returned as UTC



\$SYS\$LastWriteTime	VT_DATE	Returns the system time of the most recent write request to the PLC.  Quality is set to 0x08 if a write has not occurred since the server started up.	Read only	System time is returned as UTC
\$SYS\$StartTime	VT_DATE	Returns the startup time of the server.	Read only	System time is returned as UTC
\$SYS\$ConnStatus	VT_UI4	Returns the current known connection status to the PLC.	Read only	0 = Disconnected 1 = Connecting 2 = Connected
\$SYS\$LastConnTime	VT_DATE	Returns the system time of the most recent connection attempt to the PLC.	Read only	System time is returned as UTC
\$SYS\$LastDisconnTime	VT_DATE	Returns the system time of the most recent disconnection attempt to the PLC.  Quality is set to 0x08 if a disconnection has not occurred since the server started up.	Read only	System time is returned as UTC
\$SYS\$PLCType	VT_BSTR	Returns the PLC model number.	Read only	
\$SYS\$PLCFirmware	VT_BSTR	Returns the PLC firmware version.	Read only	
\$SYS\$ResetStatistics	VT_BOOL	To reset system item statistics, set this item to 1.  The server returns the value of this item to 0 when the reset operation finishes.	Read/Write	1 = True (reset statistics) 0 = False

\$SYS\$SyncDb	VT_BOOL	To force an upload of the tag database from the PLC to server, set this item to 1. The tag database will be uploaded, even if it is up to date.  The server automatically returns the value of this item to 0 when the upload operation finishes.	Read/Write	1 = True (force upload) 0 = False  The server automatically syncs up the tag database from the PLC. This system item can be used to ensure that the tag database in the server is synchronized to that in the PLC.
\$SYS\$PLCAccessLevel	VT_UI4	Returns the CPU Protection Level configured in the PLC. Refer to the Siemens TIA documentation for descriptions of each access level.	Read only	0 = Full Access including Fail Safe Access 1 = Full Access 2 = Read Access 3 = HMI Access 4 = No Access
\$SYS\$DiagnosticLogging	VT_UI4	Enable logger flags at protocol level. Example: All flags can be enabled by writing 15(0x0F) to this system item. The logger output folder is the same aacfg file. The file name is the same name as the PLC hierarchy node name.	Read/Write	0x01 = Info 0x02 = Warning 0x04 = Error 0x08 = Trace

## Data Conversion

The following table describes how the SIdirect OI Server 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 OI Server 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.
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 usable. 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.
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.

Quality Code	Quality State	Description
		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.
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.

## Item Validation

When items are added, they are not initially validated. Since SIDIrect symbolic addressing is in text format, the SIDIrect OI Server can only determine whether an item is valid by querying the PLC to check whether the item is defined, and then returning the data type. Validation begins after items are added.

If you are adding many items, performance would be negatively impacted if the query was performed immediately. This delayed validation improves performance.

The SIDirect OI Server contains a tag database in its address space that closely mirrors the tag database in the S7-1500 PLC. If the tag database in the PLC is not available, adding an item always succeeds and the data type VT\_EMPTY is sent to the PLC. Once the SIDirect OI Server successfully queries the PLC, the data type is updated to the correct type.

## Tested Hardware and Firmware Versions

Device	Description	Hardware and Firmware
S7-1500	Controller with built-in Ethernet	CPU1513 513-1AL00-0AB0 Hardware Version: 2 Firmware Version: V1.8.2
		CPU1516 516-3AN00-0AB0 Hardware Version: 2 Firmware Version: V1.8.2
	Failsafe controller with built-in Ethernet	CPU1511 511-1FK01-0AB0 Hardware Version: 2 Firmware Version: V1.8.2



# CHAPTER 4

## Troubleshooting the SIDirect OI Server

### In This Chapter

Troubleshooting Tools.....	95
Finding the SIDirect OI Server Version Number .....	95
Debugging Communications Between the SIDirect OI Server and the PLC .....	95
Diagnostics and Error Tracing.....	102

### Troubleshooting Tools

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

### Finding the SIDirect OI Server Version Number

This section describes how to find the version number of your OI Server.

#### To find the version number

1. From the **Start** menu, point to **Settings**, and click on the **Control Panel** option.
2. Click **Programs and Features**.
3. Find **Wonderware Operations Integration - Supervisory Siemens SIDirect OI Server** in the programs list and select it. The version is displayed. In some versions of Windows, you may have to click a hyperlink, **Click here for support information**. The release version of the OI Server appears in the **Support Info** dialog box.

OR

- Click on the **OI.SIDIR.1** Server node in the hierarchy-tree view. In the **Details** pane on the right you see the build version numbers of the respective SIDirect OI Server components.

OR

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

### Debugging Communications Between the SIDirect OI Server and the PLC

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

## To perform on-line diagnostics

- Select any active SIDirect OI Server on any node in the OI Server Manager.

The Diagnostics branch is visible only if the OI Server 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 OI Server. They allow detailed diagnostics of objects within the SIDirect OI Server.

---

**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 OI Servers. See the OI Server Manager user's guide for detailed explanation of the generic diagnostic information of OI Servers.

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 opens a window with complete diagnostic information for the 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:

<b>Name</b>	The OPC leaf item name without the fully qualified OPC item ID path.
<b>Client Value</b>	The last updated value to the client.
<b>Client Time</b>	The timestamp the client was updated last.
<b>Client Quality</b>	The item quality of last update.
<b>Subscr Msg</b>	The OI Server identifier for the subscription message of the item.
<b>Location</b>	The path item name, which is the OPC path part of the fully qualified item ID.
<b>Device Group</b>	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 OI Server. In the Diagnostics Structure branch, the structure view shows generic OI Server information as well as SIdirect OI Server-specific information. See the OI Server Manager user's guide for detailed explanation of the generic diagnostic information of OI Servers.

Name	Items	Errors	R/W Status	Value	Time	Quality	Messages	Device Groups	S7 Error	A&E
Legacy	0	0	R/W				0	0		
Legacy_01	0	0	R/W				0	0		
Legacy_02	0	0	R/W				0	0		
Symbolic	0	0	R/W				0	0		
Symbolic_01	0	0	R/W				0	0		
Symbolic_02	0	0	R/W				0	0		
Symbolic_03	0	0	R/W				0	0		
Symbolic_04	0	0	R/W				0	0		

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

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

The SIdirect OI Server provides the following additional S7-specific columns for branches:

<b>S7 Error:</b>	The S7 error code.
<b>A&amp;E:</b>	The Alarms and Events setting. Valid values are OFF, ALARMS ON, and EVENTS ON.

The items populate the following columns:

<b>Name</b>	The item leaf name.
<b>R/W Status</b>	R: Item was updated. W: No writes are pending on the item.
<b>Value</b>	The last protocol value.
<b>Time</b>	The last protocol time.
<b>Quality</b>	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 OI Server information as well as SIDirect OI Server-specific information. Refer to the OI Server Manager user's guide for detailed explanation of the generic diagnostic information pertaining to the OI Servers.

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

<b>Type</b>	The Demand Read and Demand Write icons showing the state of the transaction.
<b>ID</b>	The unique identifier for the transaction.
<b>Items</b>	The number of items.
<b>Status</b>	The status of the transaction.
<b>Start</b>	The time the transaction starts.
<b>End</b>	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 an OI Server. This generic information is not specific to the SIDirect OI Server. See the OI Server Manager user's guide for detailed explanation of the generic diagnostic information pertaining to OI Servers.

## Message View

The message view shows generic and S7-specific information of the SIDirect OI Server. See the OI Server Manager user's guide for detailed explanation of the generic diagnostic information of OI Servers.

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

<b>Msg ID</b>	The message unique identifier.
<b>Items</b>	The number of items.
<b>Errors</b>	The number of items with non-good quality.
<b>Status</b>	The status of the message.

The following is the SIdirect OI Server-specific diagnostic information:

<b>S7 Msg ID</b>	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
<b>PDU size</b>	<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.
<b>Message error</b>	The S7-message communications error code.

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

<b>Name:</b>	0100280A
<b>Active Items:</b>	462
<b>Error Items:</b>	0
<b>Status:</b>	IDLE
<b>S7 Msg ID</b>	P0
<b>PDU Size (resp/req)</b>	480/24
<b>Msg Error:</b>	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 OI Server internal message identifier.
Location	The OPC path of fully qualified item ID.

The following is the SIdirect OI Server-specific diagnostic information:

<b>S7 Msg ID</b>	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:	PORT.Symbolic...
Read status:	complete
Write status:	complete
Value:	127
Type:	VT_U11 - unsigned c...
Time:	Monday, August 10, 1015
Quality:	GOOD: Nonspecific....
S7 Msg ID:	P2
Item Error:	OK

## Devices Groups

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

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

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

For example:

Device Group:	PLC1
---------------	------

Update Interval:	1000
Number of Items:	10234
Location:	Port.Symbolic...
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 OI Server uses the standard diagnostic information provided by the OI Server (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

The SIdirect OI Server provides diagnostics in the following areas:

- *Communications Processor Diagnostics* on page 102
- *S7 Communications Processor Diagnostics* on page 102
- *Items Diagnostics* on page 102
- *Messages Diagnostics* on page 103
- *Device Group Diagnostics* on page 103

## Communications Processor Diagnostics

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

## S7 Communications Processor Diagnostics

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

## Items Diagnostics

To the IO Server-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 Diagnostics

Four types of messages with different diagnostics appear.

Variable Service Message with:

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

In addition to the standard OI Server 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 OI Server information text plus the message ID, PDU size, and error text description.

## Device Group Diagnostics

The device group diagnostics window shows the OI Server-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.

## OI Server (DASTrace) Diagnostic Messages

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

The following table is not a complete list of messages that are shown in the Logger when DASTrace is enabled. Other diagnostic messages are shown, as well.

---

**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 is set 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.

Logger Message	Explanation	Probable Cause	Solution
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 '%s' dropped, restarting connection	The socket connection with the PLC fails due to a given error code: "The PLC initiated termination of the Socket Connection."	The PLC is not responding to the Connect request from the OI Server.	The SIDIrect OI Server will attempt to automatically reconnect. No user action is required.
Connection with '%s' dropped while pending, going to slow poll	The socket connection with the PLC fails due to a given error code: "The PLC initiated termination of the Socket Connection."	The PLC has been disconnected and/or has faulted.	Check the PLC connection and clear any outstanding PLC faults.
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 OI Server 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 OI Server Logger Flags

The following logger flags are specific to the SIDIrect OI Server.



- **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.
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 OI Server error and warning messages, S7 Trace messages and generic OI Server error codes are supported. Use these messages together with the OI Server Manager Diagnostic root data to troubleshoot SIDirect OI Server 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.

## OI Server Error Messages

The following list shows error messages produced by the OI Server that are logged to the Log Viewer with the DASProtFail log flags.

Logger Message	Explanation	Probable Cause	Solution
<b>DASProtFail Messages</b>			
Internal Error: CloVariant::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 (S7 Type 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.
(%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.

<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
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 configuration or communications error occurred.	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.
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.

<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
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.
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.

<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
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 configuration or communications error occurred.	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 configuration or communications error occurred.	Check the PLC configuration/connection and resource limitation, particularly the ones related to the cyclic service.
(%d): s7_cycl_read_start_req (orderid=%d): [0]	An error occurs when sending the S7 cyclic read start request message.	A configuration or communications error occurred.	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 configuration or communications error occurred.	Check the PLC configuration or connection.

<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
ERROR in S7_CYCL_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, OI Server 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 OI Server 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 OI Server configuration error.	Turn on CYCLIC_TRACE in the Wonderware Logger to obtain additional trace information. If the problem persists, contact Wonderware Technical Support.
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 OI Server/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 (480 bytes).
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 (480 bytes).

<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
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.
%s 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.
%s 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 OI Server 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 configuration or communications error occurred.	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 configuration or communications error occurred.	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.
Can't connect	There is an error establishing a connection.	A configuration or communications error occurred.	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.

<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
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 OI Server discards the message associated with it.	If the problem persists, contact the PLC vendor.
Internal HEX ASCII conversion buffer overflow	The server cannot allocate memory for the HEX ASCII 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 occurred.	Verify the network connectivity to the PLC. Turn on CONNECTION_TRACE, DASSend, and DASReceive in the Wonderware Logger to obtain additional diagnostic information.
(%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 configuration or communications error occurred.	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.



Logger Message	Explanation	Probable Cause	Solution
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 from the PLC.	The PLC reports a cyclic response that is not requested by the OI Server.	Check with the PLC vendor.
%s 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 OI Server 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 OI Server. For more information about trace messages, see *SIDirect OI Server Logger Flags* on page 104.

Logger Message	Explanation	Probable Cause	Solution
<b>CONNECTION_TRACE Messages</b>			
(%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 OI Server closes the connection with the PLC. The parameters associated with the PLC connection are listed.	N/A	N/A
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: %s 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 OI Server closes the connection with the PLC successfully.	The connection is closed by either the server or the PLC	N/A

<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
s7_get_initiate_cnf cnf amq called: %d	INFO: PLC agrees on this parameter with the OI Server. (cnf_amq_called)	N/A	N/A
s7_get_initiate_cnf cnf amq calling: %d	INFO: PLC agrees on this parameter with the OI Server. (cnf_amq_calling)	N/A	N/A
s7_get_initiate_cnf cnf pdu size: %d	INFO: PLC agrees on this parameter with the OI Server. (cnf_pdu_size). The PDU size is returned.	N/A	N/A
s7_get_initiate_cnf ind amq called: %d	INFO: This is the OI Server side negotiation parameter with the PLC. (ind_amq_called)	N/A	N/A
s7_get_initiate_cnf ind amq calling: %d	INFO: This is the OI Server 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 OI Server 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

<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
%s Set state of connection to CLOSED	The SIdirect OI Server 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.
%s 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.
%s Set state of connection to OPEN	INFO: The server opens a connection with the PLC.	N/A	N/A
%s Set state of connection to PENDING	INFO: The server is awaiting the response from the PLC for the Connection Request.	N/A	N/A
%s 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.
<b>POLL_TRACE Messages</b>			
%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.

<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
add pltem (%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
remove pltem (%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 HEX ASCII):	INFO: The server logs the information about the read value received from the PLC in the logger.	N/A	N/A
<b>POKE_TRACE Messages</b>			
%s dumping s7 objects (containing %d items)	INFO: Dumping the information about the S7Infos in the logger.	N/A	N/A

<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
%s: result=%d var_length=%d value (in HEX ASCII)	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 pltem (%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
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 pltem (%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

<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
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
<b>CYCLIC_TRACE Messages</b>			
%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
(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

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 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 pltem (%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
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_conf (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 HEX ASCII)	INFO: Logging the value received from the PLC for the cyclic message	N/A	N/A
<b>BLOCK_TRACE Messages</b>			
(0): s7_brcv_init (r_id=%d) [0]	INFO: The block receive initiation request is successful.	N/A	N/A



<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
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
<b>ALARMS_AND_EVENTS_TRACE Messages</b>			
%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

Logger Message	Explanation	Probable Cause	Solution
(%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
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

## OI Server Error Codes

The following table lists the Wonderware OI Server 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.

Code	Error Message	Description
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.

### OI Server Protocol Warnings

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

Logger Message	Explanation	Probable Cause	Solution
<b>DASProtWarn Messages</b>			
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/pMsg=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.

Logger Message	Explanation	Probable Cause	Solution
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.
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 OI Server 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 OI Server 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 OI Server 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 OI Server 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 OI Server user's guide for the correct range of values.

<b>Logger Message</b>	<b>Explanation</b>	<b>Probable Cause</b>	<b>Solution</b>
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 OI Server 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 OI Server 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 OI Server 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 OI Server 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.



# Index

**No index entries found.**