AVEVA™ Communication Drivers Pack – General Electric - GESRTP Driver

User Guide

Publication date: Tuesday, May 9, 2023

Publication ID: 868878

Contact Information

AVEVA Group Limited
High Cross
Madingley Road
Cambridge
CB3 0HB. UK

https://sw.aveva.com/

For information on how to contact sales and customer training, see https://sw.aveva.com/contact.

For information on how to contact technical support, see https://sw.aveva.com/support.

To access the AVEVA Knowledge and Support center, visit https://softwaresupport.aveva.com.

AVEVA

Contents

Chapter 1

Introduction to the GESRTP Communication Driver

About the GESRTP Communication Driver

The GESRTP (GE Service Request Transfer Protocol) Communication Driver is a Microsoft® Windows® application program that acts as a communications protocol server. It provides other Windows application programs with access to data within the GE family of PLCs. The GESRTP Communication Driver provides access to a GE PLC through a standard Ethernet network interface card in the computer.

The GESRTP Communication Driver is primarily intended for use with the InTouch® HMI software. However, the Communication Driver can be used by any Microsoft Windows program capable of acting as a DDE, FastDDE, SuiteLink™, or OPC client. It supports the GE family of hardware and firmware listed in Tested GE Hardware.

The GESRTP Communication Driver is hosted by the OI Server Manager, a Microsoft Management Console (MMC) snap-in. Many high-level functions and user-interface elements of the OI Server Manager are universal to all Communication Drivers. Only the documentation for the OI Server Manager contains descriptions of those universal functions/UI elements.

Reading the documentation for both the MMC and the OI Server Manager is critical to understanding this Communication Driver documentation. To read the documentation about the MMC and OI Server Manager, click the Help topics on the MMC **Help** menu. Both the MMC and Communication Drivers Pack Help are opened.

Note: The GESRTP Communication Driver supports multiple network interface cards in the computer. The network card that the server uses depends on the operating system.

Supported Communication Protocols

The GESRTP Communication Driver communicates with clients and PLCs using different communication protocols. The Communication Driver uses application protocols such as OPC, DDE, SuiteLink, and PCS to communicate with the clients, and TCP bus protocols over the Ethernet to communicate with the PLCs.

For more information refer to the "Support Client Protocols" section of the AVEVA Communication Drivers Pack Help.

The GESRTP Communication Driver can have a maximum of 75 outstanding OPC/DDE/SuiteLink/PCS transactions.

Device Communications Protocols

The GESRTP Communication Driver uses only the GE Service Request Transfer Protocol, which is a TCP bus communications protocol over Ethernet, to communicate with the GE family of controllers.

Accessing Items via the Communication Driver

The method for accessing items through the Communication Driver depends on the communications protocol used. This section describes accessing items using the OPC and DDE/SuiteLink communications protocols.

Accessing Items Using the OPC Communications Protocol

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

The node name and device group are optional. A fully qualified OPC item name (ItemID) is composed of the link name and item name. All other characteristics are specified through separate Communication Driver means.

To access an OPC item, the OPC client connects to the Communication Driver, either in-process or out-of-process, and creates an OPC group defining the data-acquisition properties for the collection of items to be added. OPC groups can be either public or private. Public OPC groups are shared across multiple clients. Private OPC groups are local to a single client.

Note: The Communication Drivers only support private OPC groups.

Optionally a device group, which indicates the access path to the items for Read/Write, can be specified from the Communication Driver. The following briefly describes each characteristic of the OPC protocol.

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

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

For this Communication Driver, the base instance name is OI.GESRTP.3, and the cloned instance name would be OI.GESRTP_<cloned name>.3. The version independent ProgID is OI.GESRTP.

group name
The OPC group created from the client for logically organizing a collection of items with the same data-acquisition properties between the client and the server, such as update rate.

device group

Meaningful names configured in the Communication Driver under a specific controller for the common custom attributes between the Communication Driver and the device, such as update interval.

If not specified from the client, the default device group using the global-configuration attribute values from the Communication Driver is assumed.

Functionally, a device group is equivalent to an access path (optional).

link name
The set of hierarchy node names separated by delimiters. Link names represent the specific devices on a communications path link from the hierarchy root to a specific controller as configured for this Communication Driver under the OI Server Manager.

item name
A specific data element, the leaf of the hierarchy tree of this Communication Driver, within the specified group. For example, when using this Communication Driver, an item can be a relay, timer, counter, register, and so on, in the controller.

Connecting to the Communication Driver from an OPC Client

The GESRTP Communication Driver always runs out-of-process when instantiated, whether directly in the OI Server Manager or through OPC.

Accessing Items Using the DDE/SuiteLinkCommunications Protocol

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

The following briefly describes each portion of this naming convention.

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

application name
The name of the Windows program (this Communication Driver) that accesses the data element. For data coming from or going to the GE devices via the DDE/SuiteLink plugin of this Communication Driver, the base instance name is GESRTP, and cloned instance names are GESRTP_<cloned name>.

topic name
Meaningful names configured in the Communication Driver to identify specific devices. These names are used as the topic names in all conversations with that device.
For example, **FASTPOLL**.

Topic names map to a device group defined in the Communication Driver.

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

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

For more information on item names, see Item Naming.

Demo Mode

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

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

This process is repeated for 120 minutes, after which the Communication Driver stops updating read/write on all device items. Read from cache is allowed, but all non-system data receives Bad quality status. The GESRTP Communication Driver continues to request a license and clients continue to function normally. For example, you can still add or remove an item, but its quality is set to Bad until a license is obtained.

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

If you subsequently add a license to the License Manager, the Communication Driver logs a message acknowledging the license, switches out of Demo mode, and runs normally. After a Communication Driver obtains a valid license, it no longer checks for a license. If your license expires while the Communication Driver is running, the Communication Driver continues to function normally until it is stopped.

After a Communication Driver obtains a valid license, it no longer checks for a license. If your license expires, your Communication Driver stops functioning. This condition is not logged until the next restart of the Communication Driver.

Windows Firewall Considerations

If the Communication Driver runs on a computer with a firewall enabled, there is a list of application names or port numbers that must be put in the firewall exception list so the Communication Driver OI Server can function correctly.

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

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

- GESRTP.exe
- aaLogger.exe
- DASAgent.exe
- dllhost.exe
- mmc.exe
- OPCEnum.exe

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

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

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

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

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

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

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

Chapter 2

Configuring the GESRTP Communication Driver

Workflow for Setting up the GESRTP Communication Driver

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

1. Locate the Communication Driver in the Operations Control Management Console (OCMC). See Configuring your GESRTP Communication Driver.
2. Configure the global parameters. See Configuring Global Parameters in the Communication Drivers Pack Help.
3. Add a port. See the GESRTP_PORT Connection.
4. Add and configure ports and devices. See GEFANUC_PLC Connection.
5. Add one more device groups. See Device Group Definitions.
6. Add device items. See Device Item Definitions.
7. Activate the Communication Driver. See Preparing the GESRTP Communication Driver.
8. Access data from the client.
9. Troubleshoot any problems. See Troubleshooting the GESRTP Communication Driver.

Preparing the GESRTP Communication Driver

If you are familiar with Communication Drivers, follow the instructions in this section to prepare the GESRTP Communication Driver for use. If you are not familiar with Communication Drivers, follow the detailed procedures in Configuring the GESRTP OI Server.

Before you can configure the GESRTP Communication Driver, you need to install it and run it. There are no default values for security settings. Be sure to note the user name and password you use during the install.

To prepare the GESRTP Communication Driver

1. The GESRTP Communication Driver is installed along with the Communication Drivers Pack. It is a selectable option during the Communication Drivers Pack installation.
2. Start the Operations Control Management Console (From the Windows **Start** menu, point to **Programs**, **AVEVA**, and then click the **Operations Control Management Console** icon).
3. In the OI Server Manager tree, under the Local node, navigate to the GESRTP in the OCMC. For general information about working in this snap-in environment, see the Communication Drivers Pack Help.

   Under the **Local** node, the Communication Driver base instance name is **OI.GESRTP.1.** The version number may vary.
4. Determine the hierarchical structure of the network/PLC environment to which you plan to connect.

   The GESRTP Communication Driver uses a single-tier hierarchy for modeling the GEFANUC_PLC objects and a custom device group configuration.
5. Configure the new GESRTP Communication Driver.

   When you select the **Configuration** object in the console tree, the **Global Parameters** configuration view for the GESRTP Communication Driver opens. Configure all other global parameters as required for the GESRTP Communication Driver. For more information about the **Global Parameters** dialog box, including descriptions of the different poke modes, see "Configuring Global Parameters" in the Communication Drivers Pack Help. Any global parameters that appear dimmed are not available.
6. Optionally create device groups and device items for each logical end-point object.

   Important: When the Communication Driver or any of its configuration views is selected and you open multiple instances of the OI Server Manager, the OI Server Manager places the configuration views from the subsequent instances of the same Communication Driver into read-only mode. Access to the second instance of the Communication Driver resumes after the first one has been deselected or closed. Likewise, access to the Communication Driver configuration will be unlocked for the next instance in this order.

   For step-by-step procedures on configuring device groups, see Defining Device Groups and Device Items.

   Your GESRTP Communication Driver is now ready to use. To use the Communication Driver, you must activate it.
7. Activate the GESRTP Communication Driver.See Activating/Deactivating the Communication Driver.

Configuring your GESRTP Communication Driver

Before configuring your GESRTP Communication Driver, determine the hierarchical structure of your network/PLC environment.

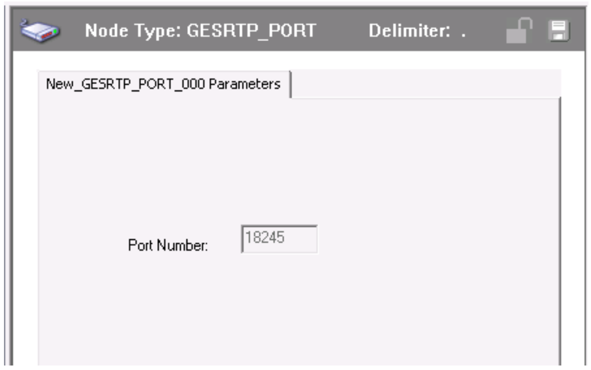
GESRTP_PORT Connection

The server-specific configuration portion of the GESRTP Communication Driver hierarchy tree under the OI Server Manager starts at the GESRTP_PORT object. This object represents a port number for the installed Communication Driver system and is used by the GESRTP Communication Driver to communicate with PLCs lower in the hierarchy tree.

You can only create a single GESRTP_Port object under the **Configuration** node in the console tree.

To add a GESRTP_PORT connection to your GESRTP hierarchy

1. In the console tree, right-click **Configuration** and then click **Add GESRTP_PORT Connection**. The **New_GESRTP_PORT_000** object appears.

   Port Number: 18245

   Edit the object name to appropriately describe components of your specific hardware environment. If you do not rename the object at this time, a numeric sequencing system is applied. You can rename the hierarchy entry later.

2. Rename this object as needed.
3. In the **Port Number** box verify the port number. The port number is always 18245.

Important: If you subsequently clear your configuration hierarchy, you must create this GESRTP_PORT object by right-clicking Configuration and then clicking **Add GESRTP_PORT Connection**. An object called **New_GESRTP_PORT_000** is created. Rename as appropriate. From this point, all of the following instructions apply.

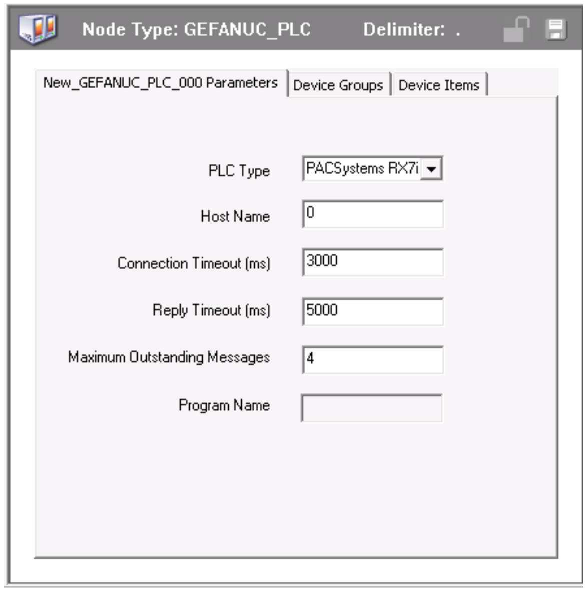
GEFANUC_PLC Connection

The GESRTP Communication Driver can connect to different GE PLCs. Each of the PLC node models the end-point of the communications path.

From the **New_GESRTP_PORT_000** branch of the Communication Driver hierarchy, create the GEFANUC_PLC object.

To add GEFANUC_PLC Connection to your GESRTP hierarchy

1. In the console tree, right-click the **New_GESRTP_PORT_000** object, and then click **Add GEFANUC_PLC Connection**. The **New_GEFANUC_PLC_000 Parameters** configuration view appear



PLC Type: PACSystems RX7i
Host Name: 0
Connection Timeout (ms): 3000
Reply Timeout (ms): 5000
Maximum Outstanding Messages: 4
Program Name:

You can create a maximum of 128 GEFANUC_PLC objects from the **New_GESRTP_PORT** branch. This maximum number can be of a single type of PLC or a combination of several types of PLCs.

2. Rename this object as needed.
3. Configure the PLC.
   - In the **PLC Type** list, click the PLC type from the list of supported GE PLCs.

     Series 90-30

     Series 90-70

     Series 90 Micro

     VersaMax

     VersaMax Micro

     VersaMax Nano

     PACSystems RX3i

     PAC Systems RX7i

     The default PLC is the GE Series 90-70 PLC. You can use this PLC-type selection to determine the PLC-variable quantities, such as the number of items in a given item name space.
   - In the **Host Name** box, type the TCP/IP address, IP address, or node name of the remote GE PLC. Type in the network address where the PLC is located or a host name if one is defined in the local hostlist.
     For example, "127.0.0.1"

     Note: The Host Name is defined in the system Host file, usually found in:
     \WINNT\system32\drivers\etc\hosts folder.

     The address must be 255 or fewer characters and cannot be blank.

- In the **Connection Timeout** box, type a value, in milliseconds, beyond which a pending request to initiate a connection times out. The allowable range is 1,000 to 600,000 milliseconds. The default value is 3,000 milliseconds.
- In the **Reply Timeout** box, type a value, in milliseconds, beyond which messages time out. The allowable range is 200 to 60,000 milliseconds. The default value is 5,000 milliseconds.

  If you decrease this value, the GESRTP Communication Driver reacts faster to a communications failure.

  For Versamax Nano and Versamax Micro PLCs that are communicating using an Ethernet-to-serial adapter, you may need to increase the reply timeout to 10 seconds or longer. A reply timeout that is set too low causes the topic to enter into slow poll mode due to hardware limitations.
- In the **Maximum Outstanding Messages** box, type the maximum number of concurrent messages that can be sent to one PLC at any time. Please refer to the communication specification of the PLC for the appropriate limit. Setting this to a higher value may increase the throughput but it will also increase the communication overhead to the PLC. Setting it to a low value will use less PLC communication resources but may also decrease the communication throughput. The allowable range is one (1) to twenty (20) messages. The default value is four (4) messages.

4. Save your changes.

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

Note: When adding a hierarchy object, the default name is in the format of **New_ObjectName_###**. ObjectName is the name of the object type and ### is a numeric value starting from "000" sequentially per hierarchy object. The link name for the OPC items is constructed by assembling the respective object names of the nodes along the hierarchy tree in the logical order, starting from the GESRTP_PORT root of this Communication Driver down to the leaf. This creates a link name that is always unique for the Communication Driver.

To use the Communication Driver, you must activate it.

To activate the Communication Driver

- In the console tree, right-click on the ProgID of the appropriate **GESTRP** instance and click **Activate Server**.

  For information about activating and deactivating the Communication Driver , see the Communication Drivers Pack Help.

Defining Poke Modes

The GESRTP Communication Driver has three poke modes for tuning the poking behavior to the PLC.

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

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

Note: Use the full optimization mode to achieve the highest performance. You need to consider your data consistency requirements with respect to poke order and folding.

Defining Device Groups and Device Items

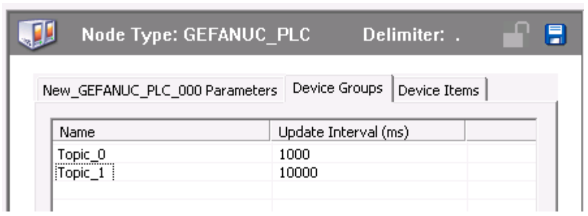
You can create new, modify, or delete device group and device item definitions for an object.

- For DDE/SuiteLink communications, one or more device group definitions must exist for each PLC that the GESRTP Communication Driver 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 GESRTP Communication Driver hierarchy tree has a **Save** button. When you modify the **Parameters**, **Device Groups** dialog box or the **Device Items** dialog box, click **Save** to implement the new modifications. If you try to open another configuration dialog box, you are prompted to save the new data to the configuration set.

Device Group Definitions

The **Device Groups** dialog box, which appears by clicking the **Device Groups** tab in the **New_GEFANUC_PLC_000 Parameters** configuration view, is the place where you create, add, delete, and define device groups. You can also configure default update intervals for the objects and edit update intervals in this dialog box.



Name	Update Interval (ms)
Topic_0	1000
Topic_1	10000

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

To create or add device groups

1. Right-click in the **Device Groups** box and click **Add**.
2. Enter a unique name up to 32 characters long for the device group.

To delete device groups

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

To make edits on device groups

- Use the **Edit** option from the **Device Groups** tab only for configuring the Communication Driver's unsolicited message handling.

  For detailed information about unsolicited message handling and configuring it, see Unsolicited Message Handling.

To configure default update intervals

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

To edit update intervals

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

  OR

  Right-click its value in the **Update Interval** column and then click **Modify Update Interval**.

  - The update interval is the frequency, in milliseconds, that the GESRTP Communication Driver acquires data from the topics associated with that device group.
  - Different topics can be polled at different rates from a PLC by defining multiple device group names for the same PLC and setting a different update interval for each device group.

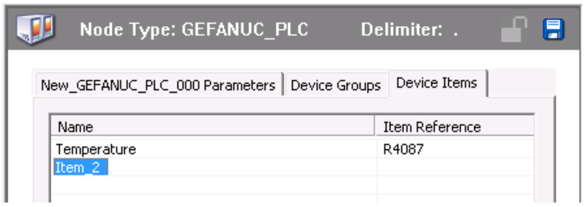
Note: When you select another part of the GESRTP Communication Driver tree hierarchy, you are prompted to save the modifications to the configuration set.

Device Item Definitions

You cannot change the predefined item syntax/name for the GE PLC. 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 "R4087" as "Temperature."

The GESRTP Communication Driver supports a maximum of 1,000 aliases.

Select the **Device Item**s tab in the OI Server Manager user interface to create new, modify, delete, export, or import device item definitions for an object. Configure the device item names using the **Device Items** dialog box, which appears when you click the **Device Items** tab in the **New_GEFANUC_PLC_000 Parameters** configuration view.



Name	Item Reference
Temperature	R4087
Item_2	

After you configure item names, the Communication Driver can perform OPC item browsing. When the Communication Driver is running and an OPC client requests item information, the configured items appear under the PLC hierarchy node. User-defined data types appear with a flat address space when browsed through OPC.

Note: The Communication Driver does not provide run-time browsing of PLC items using the SRTP protocol or the GE programming software.

To create or add device items

1. Right-click in the **Device Items** box and click **Add**.
2. In the **Name** column, type a unique item name. The maximum is 32 characters. For example, "Temperature."

3. Double-click the line in the **Item Reference** column and enter the correlated item reference for the name you created. For example, "R4087."

Note: If the name and the item reference are the same, it is only necessary to enter a name. The Communication Driver 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 the device item to be renamed and click **Rename**. Make the changes.

To delete device items

- Right-click 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 Communication Driver Item Data

fter you configure device items, you can export and import the Communication Driver item data to and from a CSV file. 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 Communication Driver item data to a CSV file

1. Right-click in the **Device Items** box and click **Export**. The **Save As** dialog box appears. The file name defaults to "PLC Hierarchyname.csv" within the default directory.
2. Accept the defaults to save the file. The file is saved as New_GEFANUC_PLC_000.csv and is editable in Microsoft Excel.

   The file contains one row for each item configured with two columns: Name and Item Reference.
3. Make your edits to the CSV file.

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

To import Communication Driver item data from a CSV file

1. Right-click in the **Device Items** box.
2. Click **Import** on the menu. The **Open** dialog box appears.
3. 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 Communication Driver is running and an OPC Client browses for items, the imported configured items appear under the PLC hierarchy node.

Scan-Based Message Handling

The Communication Drivers poll hardware devices for information. This polling is requested by one or more clients.

After a client requests a particular piece of information, the GESRTP Communication Driver creates its own request and sends that request to the hardware device. The GESRTP Communication Driver then waits for a response to its request. After the GESRTP Communication Driver 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 GESRTP Communication Driver polls a particular device for a specific piece of information in the device group (topic definition) inside the GESRTP Communication Driver. 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 seconds. However, if every item is polled at this rate, the entire system suffers due to slow response time. Therefore, compromise and set the update interval to a more reasonable value.

You could also create multiple device groups for each device, setting the update interval to different values, then assigning different items to different device groups, depending on how quickly the values change and how quickly you want to see an update of those changes.

Some items, like alarms, change very infrequently but because of their importance require very fast updates. For those kinds of items, you can set the update interval at a very small value. If you want an immediate response, set the update interval to 1.

Unsolicited Message Handling

A PLC can detect when a critical event occurs before the Communication Driver has a chance to poll for that data. If a PLC determines that a critical condition exists, it can generate a message and immediately send it to the Communication Driver. This type of message is called an unsolicited message. The GESRTP Communication Driver supports unsolicited messages from the PLCs.

The following is required to support unsolicited messaging:

- The messaging instructions are properly programmed in the PLC logic.
- The device group is appropriately configured in the Communication Driver.
- Symbolic variables cannot be specified.

To use unsolicited messaging for the GESRTP Communication Driver, you must set up unsolicited messaging from the PLCs. For more information, see the GE Automation documentation.

Note: Unsolicited messaging is only supported in the base instance of the Communication Driver.

To configure the GESRTP Communication Driver to receive unsolicited messages

1. Click on the target PLC node under the PLC branch of the Communication Driver hierarchy.
2. Select the **Device Group** tab and add a new device group or select an existing device group that needs to be configured to receive unsolicited messages.
3. Right-click on the device group name and then click **Edit**. The **Device Group Parameters** dialog box appears.
4. Select the **Support Unsolicited Messages** check box to have the Communication Driver process unsolicited messages for the device group selected.
5. Click **OK**.

Note: Because you cannot view the status of Support Unsolicited Messages check box from the Device Groups tab, proper naming of device groups which support unsolicited messages is strongly recommended.

To enhance performance in message handling, the device group is to not provide unsolicited message data. By default the **Support Unsolicited Messages** check box is selected. The setting of this check box is hot-configurable. Unsolicited message handling takes effect in the Communication Driver as soon as you save the change made in the configuration view.

To receive unsolicited messages

1. Activate the Communication Driver.
2. Under the device group set up for receiving unsolicited messages, add the items, defined in the PLC for unsolicited messages.

Archiving Configuration Sets

After you configure your Communication Driver, you can archive that specific configuration. You can archive more than one configuration set and select different configurations for different purposes.

To archive configuration sets

1. In the OI Server Manager console tree, right-click **Configuration** and then click **Archive Configuration Set**.
2. In the **Archive Configuration Set** dialog box, provide a configuration set name.
3. Click **Archive**. All current configuration values are saved to the archived set.

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

To use different configuration sets from the current one

- In the OI Server Manager console tree, right-click **Configuration**, point to **Use Another Configuration Set**, and then click on a configuration set. All parameters in the Communication Driver configuration hierarchy change to the selected configuration set.

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

Activating/Deactivating the Communication Driver

When you activate the Communication Driver, it starts communicating and accepting requests from client applications. There are three different modes of activating the Communication Driver.

1. **Activate (Auto start after reboot)**: Activates the Communication Driver. The Communication Driver is started and activated automatically when the computer starts up.
2. **Activate until reboot (Manual start after reboot)**: Activates the Communication Driver. The Communication Driver ets deactivated after a reboot, and has to be activated manually.
3. **Desktop mode (Must start from command line)**: Activates the Communication Driver from the command-line only, and not from the OCMC. This option is enabled for base instance of the Communication Driver only. For all cloned instances, this option is disabled.

   Note: Use the desktop mode activation for DDE/FastDDE communications.

To activate the Communication Driver

1. In the OI Server Manager, navigate to the Communication Driver.
   a. Expand **Operations Integration Server Manager**, expand **Default Group**, and then expand **Local**.

   b. Expand **Operations Integration Supervisory Servers**.
   c. Locate the Communication Driver you are trying to activate.
2. Right-click the base instance of the Communication Driver, and select one of the modes to activate the server.

   Selecting any one mode of activation disables all other activation options in the menu. To activate the server in any other mode, you must deactivate the server first.

To deactivate the Communication Driver

1. In the OI Server Manager, navigate to the Communication Driver.
   a. Expand **Operations Integration Server Manager**, expand **Default Group**, and then expand **Local**.
   b. Expand **Operations Integration Supervisory Servers**.
   c. Locate the Communication Driver you are trying to activate.
2. Right-click the base instance of the Communication Driver, and select **Deactivate (Must be activated to run again)**.
3. Read the warning message and click **Yes**.

   Deactivating your Communication Driver stops it from communicating with client applications.

Note: Communication Driver with active OPC clients does not stop until the last OPC client shuts down.

Working with Device Redundancy

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

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

If the Primary device fails, the Communication Driver will automatically switch over to the Secondary device. The Secondary device then becomes the active device with the failed Primary device reverting to the backup role. If the failed device recovers to good status, it will remain in the standby mode.

Understanding Runtime Behavior

The Communication Driver will start with the active device. The OI Engine 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 OI 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 OI 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 OI 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.

Refer to Communication Driver Redundant Device-Specific System Items for system items specifically associated with a Redundant Device.

Note: The Ping Item must be a valid item from the controller that has not been rejected by the server for the failover to function properly.

The Communication Driver 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.

Hot Configuration

Value changes for hot-configurable parameters take effect immediately while the Communication Driver is running. The GESRTP Communication Driver supports hot configuration for the following actions:

- Modifying global configuration parameters.
- Adding, deleting, or modifying device nodes without affecting any other device nodes, excluding the children of the modified device nodes.
- Adding, deleting, or modifying device groups including unsolicited message handling, the **Update Interval** column in the **Device Groups** tab, and device items.
- Modifying server-specific configuration parameters, except **NodeAddress** and **PLCType**, in the configuration view.

All other parameters are not hot-configurable. To make those changes take effect, reset the corresponding hierarchy. See "Hot Configuration" in the Communication Drivers Pack Help for information about how to do this.

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

Chapter 3

GESRTP Communication Driver Reference

- Item Naming
- Tested GE Hardware

Item Naming

The GESRTP Communication Driver uses an item-naming convention based on the single-letter data-type identifiers used in programming the GE PLCs, in addition to system variable addressing.

The tables in this documentation describe the item naming convention for the GESRTP Communication Driver. The ranges specified in those tables vary according to the type of the controller used.

If you advise an out-of-range variable for an item, all items in the same packet return bad quality.

Variable Names

The GESRTP Communication Driver supports PAC System variable addressing with user-friendly names.

Base Item Names

This section describes the base item names and the conditions applicable to them:

- Item names must begin with base name strings and optionally preceded by "%."
- Item names are not case-sensitive.

The following table summarizes the base name, description, data type, read/write (R/W) designation, and offset for base item names.

Base Name	Description	Data Type	Access Rights	Offset
In	Discrete inputs	VT_BOOL	R/W	n = 1-65535
On or Qn	Discrete outputs	VT_BOOL	R/W	n = 1-65535
Tn	Discrete temporary	VT_BOOL	R/W	n = 1-65535

Base Name	Description	Data Type	Access Rights	Offset
Mn	Discrete internal	VT_BOOL	R/W	n = 1-65535
Gn	Genius global data	VT_BOOL	R/W	n = 1-65535
GAn	Genius seamless A table*	VT_BOOL	R/W	n = 1-64255
GBn	Genius seamless B table*	VT_BOOL	R/W	n = 1-62975
GCn	Genius seamless C table*	VT_BOOL	R/W	n = 1-61695
GDn	Genius seamless D table*	VT_BOOL	R/W	n = 1-60415
GEn	Genius seamless E table*	VT_BOOL	R/W	n = 1-59135
W*n*	PLC WORD register	VT_UI2 or forced by suffix	R/W	n = 1-2147483647
Rn	PLC register	VT_UI2 or forced by suffix	R/W	n = 1-65535
Rn:b Similarly for data types %AI, %AO or %AQ, %P, BLOCKNAME^%L, %W	PLC register bit	VT_BOOL	Read-Only (R/O)	b = 0-15
Rn-Rm M Similarly for data types %AI, %AO or %AQ, %P, %W. For %L, the syntax is BLOCKNAME^%Ln-Lm M	String contained in consecutive bytes in the PLC registers	VT_BSTR	R/W	1. n,m = 1-65535 2. m >= n 3. For Micro and Nano PLCs, (m-n) < 100 For all other PLCs, (m-n) < 1000

Base Name	Description	Data Type	Access Rights	Offset
Rn-Rm A Similarly for data types %AI, %AO or %AQ, %P, %W. For %L, the syntax is BLOCKNAME^%Ln-Lm A	String contained in consecutive PLC registers	VT_BSTR	R/W	1. n,m = 1-65535 2. m >= n 3. For Micro and Nano PLCs, (m-n) < 100 For all other PLCs, (m-n) < 1000
AIn	Analog inputs	VT_UI2 or forced by suffix	R/W	n = 1-65535
AOn or AQn	Analog outputs	VT_UI2 or forced by suffix	R/W	n = 1-65535
Sn	System discretes	VT_BOOL	R/O	n = 1-128
SAn	System discretes	VT_BOOL	R/O	n = 1-128
SBn	System discretes	VT_BOOL	R/O	n = 1-128
SCn	System discretes	VT_BOOL	R/O	n = 1-128
BLOCKNAME^ %Ln	Sub-program block registers**	VT_UI2 or forced by suffix	R/W	BLOCKNAME = corresponding block name in the control program of the PLC. n = 1-8192
%Pn	Task memory of the PLC control program***	VT_UI2 or forced by suffix	R/W	n = 1-8192
%Wn	Retentive Bulk Memory Area, which is referenced as %W (WORD memory).	VT_UI2 or forced by suffix	R/W	*n* = 1- 2147483647 (which is a 4 byte address range/2).

* Items GA, GB, GC, GD, and GE are supported only by the Series 90-70 PLCs.

** Item %L is supported for Series 90-70 PLCs and PACSystems RX3i/7i.

*** Items %P is supported only for Series 90-70 PLCs.

If the number of values written to a string item (for example, R1-R20 M or R1-R20 A) is greater than the number of elements in the string or if the item value is out of range, then the transaction is shown as Error with a description of Write Result as Error[The value is out of range.].

Status Items

The GESRTP Communication Driver provides a number of status items. The following conditions apply to these status items:

- Optionally, '%' precedes the status items.
- Item names are not case-sensitive.

The following table summarizes the base name, description, data type, Read/Write designation, and offset for Status Items.

Base Name	**Description**	**Data Type**	**Access Rights**	**Offset**
EnSwitch	The status of the Enable switch on the PLC front panel.	VT_BOOL	Read-Only (R/O)	None
Run	The PLC run state. 0 = Stop; 1 = Run.	VT_BOOL	R/O	None
IOStatus	The Status bit that is piggybacked on the Fault table.	VT_BOOL	R/O	None
FTStatus	The Fault table status. **Note:** GE Fanuc Versa Max PLC does not support the FTStatus item. This is a GESRTP Communication Driver specific item. Use ANY_FLT, PLC system item, which gives the fault table status for any GE Fanuc PLC.	VT_BOOL	R/O	None
Interval	The update interval of the topic.	VT_UI4	R/O	None

Base Name	Description	Data Type	Access Rights	Offset

Time	The PLC Time string reads the PLC clock as a 14-digit string: SSMMHHDDMoYYWd.
	
	Where:
	SS = Seconds [00-59]
	MM = Minutes [00-59]
	HH = Hours [00-23]
	DD = Day of Month [01-31]
	Mo = Month of Year [01-12]
	YY = Year of Century [00-99]
	Wd = Day of the Week [1- Sunday, 2 - Monday… 7 - Saturday]
	
	Note: The GE Nano PLCs do not support Time items. Do not follow the above Time string format when advising Time items for Nano PLCs.
		VT_BSTR	R/O	None

OverSweep	The PLC scan fails to keep up.	VT_BOOL	R/O	None
OEMProt	The OEM Protect bit.	VT_BOOL	R/O	None
SNum_Progs	The number of programs in the PLC.	VT_UI1	R/O	None
SProg_Flags	Program flags.	VT_UI1	R/O	None
SPrgChg	Program change flags.	VT_BOOL	R/O	None
SweepTime	The PLC sweep time in increments of 100 microseconds.	VT_UI2	R/O	None
State	The current PLC state. 0 = Run I/O enabled. 1 = Run I/O disabled. 2 = Stop I/O disabled. 3 = CPU stop faulted. 4 = CPU halted. 5 = CPU suspended. 6 = stop I/O enabled.	VT_UI1	R/O	None

Base Name	Description	Data Type	Access Rights	Offset
SNPID	The PLC ID on the SNP bus. **Note:** GE PACSystems PLCs (RX3I and RX7I) do not support the SNPID item. The CPU module's configuration in PAC Systems PLC program do not have the SNPID property. Other GE Fanuc PLC's hardware modules have the SNPID property	VT_BSTR	R/O	None
NewIOFT	The new I/O Fault since last reset.	VT_BOOL	R/O	None
NewFT	The new PLC Fault since last reset.	VT_BOOL	R/O	None
ProgName	The name of the current program.	VT_BSTR	R/O	None
ProgAttach	The program attached.	VT_BOOL	R/O	None
PrivLevel	The current Privilege Level. Level-0 - (Series 90-70 PLC only.) Read and write of the PLC. Level-1 - Read any data memory. Write to memory is prohibited. The PLC cannot be started or stopped. Level 2 - Write to any data memory, except for overriding discrete I/O. The PLC can be started or stopped. PLC and I/O fault tables can be cleared. Level 3 - Write to any configuration or logic, including word-for-word changes, the addition/deletion of program logic, and the overriding of discrete I/O. Level 4 - Write to all configuration or logic. Configuration may only be written in STOP mode; logic may be written in STOP or RUN mode. Display, set, or delete passwords for any level.	VT_UI1	R/O	None
ConSweep	The constant Sweep Mode flag.	VT_BOOL	R/O	None
SY_Full	Set when the PLC Fault table fills up.	VT_BOOL	R/O	None
IO_Full	Set when the I/O Fault table fills up.	VT_BOOL	R/O	None

Base Name	Description	Data Type	Access Rights	Offset
OVR_PRE	Set when an override exists in %I, %Q, %M, or %G memory.	VT_BOOL	R/O	None
PLC_BAT	Set to indicate the bad battery in the CPU.	VT_BOOL	R/O	None
USR_SW	Set to indicate the CPU mode switch. 1: Run/On 0: Stop/Off **Note:** Only the GE Fanuc Versa Max PLC supports the USR_SW item.	VT_BOOL	R/O	None
CFG_MM	Set when a configuration mismatch is detected during a power-up or a configuration store.	VT_BOOL	R/O	None
HRD_CPU	Set when diagnostics detects problems with the CPU hardware.	VT_BOOL	R/O	None
LOW_BAT	Set when a low battery fault occurs.	VT_BOOL	R/O	None
BAD_PWD	Set when a password violation occurs.	VT_BOOL	R/O	None
SFT_CPU	Set when the CPU detects an error in the CPU operating system.	VT_BOOL	R/O	None
ANY_FLT	Set when any fault occurs. 1 = Fault is present (either PLC fault or I/O fault) 0 = No fault	VT_BOOL	R/O	None
SY_FLT	Set when any fault occurs and results in an entry to the PLC Fault table.	VT_BOOL	R/O	None
IO_FLT	Set when any fault occurs and results in an entry to the I/O Fault table.	VT_BOOL	R/O	None
SY_PRES	Set as long as there is at least one entry in the PLC Fault table.	VT_BOOL	R/O	None
IO_PRES	Set as long as there is at least one entry in the I/O Fault table.	VT_BOOL	R/O	None
HRD_FLT	Set when a hardware fault occurs.	VT_BOOL	R/O	None
SFT_FLT	Set when a software fault occurs.	VT_BOOL	R/O	None

Base Name	Description	Data Type	Access Rights	Offset
PLCFTCOUNT	The number of faults present in the PLC Fault table.	VT_UI1	R/O	None
IOFTCOUNT	The number of faults present in the I/O Fault table.	VT_UI1	R/O	None
PLCFT_n	The fault entry n of the PLC Fault table. The maximum number of PLC fault entries is 16.	VT_BSTR	R/O	None
IOFT_n	The fault entry n of the I/O Fault table. The maximum number of I/O fault entries is 32.	VT_BSTR	R/O	None

Symbolic Variables

Symbolic variables represent named ranges in the memory of the GE PLCs, which you can define without knowing the addresses of specific memory registers. You can also use them to access pre-defined tags in the PLC.

Important: Array and symbolic variable addressing is only supported in subscribing data from the PAC Rx3i and PAC Rx7i controllers. Because of protocol limitations, array or symbolic variable addressing is not supported from other GE controllers.

For example, if you know there is a tank pressure sensor connected to a PLC, you can access the data from its tag as "#TankPressure" rather than having to know the exact memory register address for that data.

Symbolic variables can be addressed with the following syntax:

```
#<var_name> <ItemSuffix>
```

(where "#" and <ItemSuffix> are optional).

- Prefix '#' - This identifies the item as a symbolic variable. If the PLC has a symbolic item with the same name as a register (such as R3000) then the '#' is required.

  If '#' is not provided, then the GESRTP Communication Driver will first try to resolve the item as an IO or memory register. If it is not a valid syntax for an IO or memory register, then the Communication Driver will create the Symbolic Item.

- <Item Suffix> - This identifies the data type, when the item is created in the Communication Driver. Combination of suffixes is not applicable.

  If <Item Suffix> is not supplied, the DA Server will create the item with delayed item validation. The data type at the time of item creation will be VT_EMPTY and the actual PLC data type will be determined later. For more information on data types and suffixes, see Suffixes to Explicitly Control I/O Variables and Data Types for Registers.

Examples:

#TankLevel ARRAY(S)

This creates an array named "TankLevel," with the canonical data type of VT_I2 (2 byte integer).

#R3000 W

This creates a symbolic variable named "R3000," with the canonical data type of VT_UI2 (2 byte unsigned integer).

TankLevel

This creates a symbolic variable item named "TankLevel." The data type will be determined when the Communication Driver attempts to access the item on the PLC.

Reference Variables

A reference variable is a variable that you can use in logic to refer to another register memory or array elements.

For example, the symbolic variable 'TankLevel' is reference to the register memory '%R100'. In this case when user advises 'TankLevel' it will internally refer to memory address of 'R100'.

The syntax for reference variables is similar to the symbolic variables.

Syntax:

```
#<var_name> <ItemSuffix>
```

(where "#" and <ItemSuffix> are optional).

Example:

#TankLevel S

Where 'TankLevel' is an reference to register memory 'R100'.

User-Defined Data Types

A user-defined data type (UDT) is a structured data type consisting of elements of other selected data types.

The data type of each element of a UDT can be one of the following:

- A simple data type, except string.
- Another UDT.
- An array of a a simple data type.
- An array of UDT.

Note: A UDT cannot be nested within itself.

UDT Syntax:

#<UDT Instance>.<UDT Element> <ItemSuffix>

(where "#" and <ItemSuffix> are optional).

Arrays

Arrays are a collection of variables under a single name, ordered by a numeric index. Arrays can be one dimensional or two dimensional, with each dimension value separated by comma, if necessary.

Arrays are referenced as a whole, using the array item syntax, or by individual element. If the whole array is requested, the length of the requested data from the array must not exceed the maximum PDU size limit (2000 bytes).

Array Item Syntax

`#<PLCVariableName> ARRAY(<DatatypeSuffix>)`

Where <DatatypeSuffix> is one of these suffixes: BT, S, W, L, DW, BY, F, LF, M.

The array may be passed to the variable as a series of values separated by spaces, tabs or carriage return/line feeds. It will be returned by the PLC as a series of value separated by carriage returns/line feeds.

To poke new values to the array, you cannot poke a partial array, but must write the full length array string.

Important: Array and symbolic variable addressing is only supported in subscribing data from the PAC Rx3i and PAC Rx7i controllers. Because of protocol limitations, array or symbolic variable addressing is not supported from other GE controllers.

For example, if the original array value was "1 2 3 4 5" and you needed to replace the "3 4" with "6 7," you would need to poke "1 2 6 7 5."

Examples:

#TankArray ARRAY(L)

This creates an array named "TankArray," with the canonical data type of VT_I4 (4 byte integer).

#R3000 ARRAY(W)

This creates a symbolic variable named "R3000," with the canonical data type of VT_UI2 (2 byte unsigned integer).

Array Element Item Syntax

`#<PLCVariableName>[n] <ItemSuffix>`
Where n is the element number from the array.

Examples:

#TankArray[2] L

This creates array item 2 in the array "TankArray," which has the canonical data type of VT_I4 (4 byte integer).

#R3000[2,3] W

This creates an array item in row 2, column 3 of the two-dimensional array "R3000," which has the canonical data type of VT_UI2 (2 byte unsigned integer).

Item Name Syntax

The item syntax for <PLCVariableName> is not validated by the GESRTP Communication Driver for invalid characters or length. The variable name will be sent to the PLC as it is and the PLC will reject the item if it is invalid.

Suffixes to Explicitly Control I/O Variables and Data Types for Registers

This section describes the suffixes used to explicitly control data type. The following conditions apply to these suffixes:

- You can universally apply suffixes to all supported item names.
- You can either have a single suffix or a combination of the suffix and the A suffix. The A and M suffixes cannot be used together.

- Each suffix applied must be preceded by a space: R77-R100 A (BCD)

The following sections summarize the suffix group, suffix, restrictions, description, and result for Suffixes to Explicitly Control Data Type.

- A Suffix
- B or BCD or (BCD) Suffix
- BT Suffix
- BY Suffix
- DW Suffix
- F Suffix
- L Suffix
- LF Suffix
- M Suffix
- S Suffix
- W Suffix

Supported Data Types

Item Suffix	**PLC Data Type**	**Variant Type**
B or BCD or (BCD)	BCD data	VT_UI2
BT	BOOL (Bit)	VT_BOOL
BY	BYTE	VT_UI1
DW	DWORD	VT_UI4
F	REAL	VT_R4
L	DINT	VT_I4
LF	LREAL	VT_R8
M	STRING	VT_BSTR
S	INT	VT_I2
W	UINT, WORD	VT_UI2
ARRAY(sfx)	Array	Data type indicated in parentheses.

A Suffix

You can interpret consecutive PLC register values as an ASCII string of values separated by spaces by specifying a register range followed by a blank space and an A. For example, R501-R510 A.

This item name indicates that registers R501 through R510 should be read and written as a unit. This technique is useful to load a group of registers with a new set of control values. For example, a recipe.

You can assume registers to be unsigned, unless otherwise specified.

R100 A is the same as R100-R101 A.

In addition to the A, by further suffixing a blank space and an S (signed), or L (long integer), or F (floating point), or (BCD), the string can be interpreted as a string of signed / long / floating point / BCD.

The resulting data type is a VT_BSTR.

For example,

R1-R6 A (BCD)

P1-P10 A

AI1-AI10 A

R1000-R1099 A S

WIDGETS^%L1-L20 A S

AO1-AO10 A.

Example:

R1 = 54, R2 = 55, R3 = 69, R4=72, R5 = 73

R1-R5 A = 54 55 69 72 73

The following point names support register arrays (blocks): %AI, %AO or %AQ, %P, %R and BLOCKNAME^%L, %W.

When you write or poke a new value for the Register Array from the client, it must be in the form of a character string containing a value for each register. The register values can be separated by any combination of commas, tabs, spaces, carriage returns, and line feeds.

For example, if the item is R1-R6 A, you poke the following values:1,2,3,4,5,6 or
1<tab>2<tab>3<tab>4<tab>5<tab>6 or 1 2 3 4 5 6, and so on.

When the server returns a new value for a Register Array to the client, it is in the form a character string containing a value for each register separated by carriage return and line feed.

For example, if the item is R1-R6 A, the value returned is 50<cr><lf> 17<cr><lf> 0<cr><lf> 5<cr><lf> 1007<cr><lf> 20<cr><lf>.

Supported base items are: %AI, %AO or %AQ, %P, %R, %L, %W

Note: For Micro and Nano PLCs, a maximum of 100 elements are allowed for items with suffix A. For all other PLCs, the maximum is 1000.

B or BCD or (BCD) Suffix

You can interpret a register as a BCD (Binary Coded Decimal) by adding a blank space and BCD following the item name. For example, R4087 (BCD) or R4087 B or R4087 BCD.

This causes the server to convert the register contents to BCD before sending it to the client application.

Supported base items are: %AI, %AO or %AQ, %P, %R, %L, %W

BT Suffix

All items with data format with suffix "BT" (Bit) will read and write a single bit in the PLC memory. The data type is always VT_BOOL for read/write.

BY Suffix

You can interpret the specified register as a BYTE, an 8-bit unsigned integer value. This is lower byte in the register of 2 bytes.

For Example: R20 BY

This notation causes server to treat lower byte of R20 as 8-bit unsigned integer.

DW Suffix

You can interpret consecutive PLC register values as a DWORD, or unsigned long integer, by adding a blank space then DW after the item name.

For example: %R95 DW

This notation causes the server to treat registers R95 and R96 as an unsigned 32-bit number with R95 being the least significant half.

F Suffix

You can interpret a pair of analog or block registers as a floating point number by adding a blank space and an F to the item name for the lower-numbered register of the pair. For example, R1001 F.

This notation causes the server to treat R1001 and R1002 as IEEE 32-bit floating point numbers. The resulting type is a VT_R4. This suffix can only be used with suffix A.

Supported base items are: %AI, %AO or %AQ, %P, %R, %L, %W

L Suffix

You can interpret a pair of analog or block registers as a long integer or double-precision integer by adding a blank space and an L after the item name. For example, WIDGETS^%L95 L.

This notation causes the server to treat L95 and L96 in sub-program block WIDGETS as a signed 32-bit number with L95 being the least significant half. The resulting type is a VT_I4.

This suffix can only be used with suffix A.

Supported base items are: %AI, %AO or %AQ, %P, %R, %L, %W

LF Suffix

You can interpret four consecutive registers, analog, or block register as an LREAL, or double precision floating point number, by adding a blank space then an LF to the item name for lower numbered register of the pair.

For example: R1001 LF

This notation causes the server to treat R1001, R1002, R1003 and R1004 as an IEEE 64-bit double-precision number.

M Suffix

You can treat a series of consecutive analog or block registers as an ASCII character string by adding a blank space and an M following the item name. For example, R101-R150 M.

This item name indicates that registers R101 through R150 contain a string of 100 ASCII characters.

Each register contains two characters with the low-order byte first. A zero byte in any register is treated as the end of the string. A single register followed by the M suffix is treated as a 4-byte long register string. For example, assume the following values are in the PLC: R1 = 54, R2 = 55, R3 = 69, R4=72, R5 = 73, R1-R5 M = 67EHI. For example, R101 M is the same as R101-R102 M.

For a single register, specify R101-R101 M.

The server reads R100 and R101 register values in ASCII with a space in between.

The resulting type is a VT_BSTR.

This suffix cannot be used with any other suffix.

Supported base items are: %AI, %AO or %AQ, %P, %R, %L, %W

Note: For Micro and Nano PLCs, a maximum of 100 elements are allowed for items with suffix M. For all other PLCs, the maximum is 1000.

S Suffix

Reads the specified register(s) as a signed value. For example R001 S. The resulting type is a VT_I2.

This suffix can only be used with suffix A.

Supported base items are: %AI, %AO or %AQ, %P, %R, %L, %W

W Suffix

You can interpret consecutive PLC register values as a WORD, or unsigned 16-bit integer, by adding a blank space then W after the item name.

For example: %R95 W

Conversions and Suffixes of Items

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

All items marked "Forced by suffix" in the Data Type column of the Status Items and Base Item Names tables are affected by any of these suffixes. For acceptable combinations of suffixes, see the table in Suffixes to Explicitly Control I/O Variables and Data Types for Registers.

Suffix A

All items with data format integer and with suffix A represent an ASCII string of values stored in consecutive PLC registers. The values are separated by spaces.

By default data format of each element in the string is an integer and is forced by the additional suffix applied on it (F-float, S-signed, L-long, B-bcd). The data type is always VT_BSTR for read/write.

Suffix B or BCD or (BCD)

All items with suffix B or BCD are converted from the BCD format into the integer and back.

Suffix BT

All items with data format with suffix "BT" (Bit) will read and write a single bit in the PLC memory. The data type is always VT_BOOL for read/write.

Suffix BY

All items with data format with suffix "BY" will read and write lower Byte in PLC register. The data type is always VT_UI1 for read/write.

Suffix DW

All items with data format and with suffix "DW" will read and write twice the number of PLC registers as their normal format would. The data type is always VT_UI4 for read/write.

Suffix F

All items with suffix F are converted to type VT_R4 on Read and require type VT_R4 on Write.

Suffix L

All items with data format integer and with suffix L read and write twice the number of PLC registers as their normal format does.

Suffix LF

All items with data format with suffix "LF" will read and write 4 PLC registers as their normal format would. The data type is always VT_R8 for read/write.

Suffix M

All items with suffix M are converted to type VT_BSTR (consecutive bytes of the specified registers) on Read and require type VT_BSTR on Write.

For array items, the values are separated by spaces. All items with data format integer and with suffix M represent an ASCII string of values stored in consecutive PLC registers or array elements in Symbolic Variable.

Suffix S

All items with suffix S are forced to a signed representation on Read and accept signed values on Write.

Suffix W

All items with data format and with suffix "W" will read and write twice the number of PLC registers as their normal format would. The data type is always VT_UI2 for read/write.

Communication Driver Standard System Items

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

No Communication Driver-specific system items are provided in this GESRTP Communication Driver.

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

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

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

```
<GESRTP_Port name>.<GEFANUC_PLC>.$SYS$Status
```

Example:

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

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

Note: For DDE/SuiteLink clients, $SYS$Status always comes from the leaf level of a Communication Driver hierarchy branch, which is defined by the unique device group. For OPC clients, $SYS$Status can be accessed at all hierarchy levels. $SYS$Status at the root level of the whole hierarchy tree is always good, as it represents the quality status of the local computer itself. For practical application, OPC clients should reference $SYS$Status at hierarchy levels other than the root.

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

All system items follow the same naming convention:

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

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

Communication Driver Global System Item

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

System Item Name (Type)	Type/Access Rights	Description

$SYS$Licensed	Boolean/Read	Binary status indication of the existence of a valid license for the OI Serve If FALSE, this item causes the OI Server to stop updating existing tags, to r activation of new tags, and to reject write requests in addition to setting c all items to BAD. If TRUE, the OI Server functions as configured. All instances have the same value.
$SYS$ReadOnly	Boolean/Read	Binary status indication of the Read Only state of an OI Server. If TRUE, the Read/Write access of all items are overridden to read only an be written. If an item is written, a line is logged in the SMC Logger and the request is rejected. If FALSE, the OI Server items are read/write or read or according to their individual configurations.

Communication Driver Device-Specific System Items

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

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

System Item Name (Type)	Type/ Access Rights	Description	Values
$SYS$ErrorText	String/ Read	Detailed error string of the communications state of the device. The device group (OPC access path/topic) does not affect the value.	Descriptive text for the communications state corresponding to the error code.
$SYS$StoreSettings	Integer/ Read/ Write	Make the temporary update interval changes via the $SYS$UpdateInterval item permanent. If the client pokes a value of 1 into this system item, the currently set update interval is written to the servers configuration file. The value of this system item clears to 0 after being set if the configuration file write is successful. If the write fails, then the value is set to -1. If the update interval is changed via the $SYS$UpdateInterval item and this item is not poked to 1, the Communication Driver uses the original update interval for that topic the next time it is started. Reading the item always provides 0. Read/Write values are persisted only if you set this system item. The values other than this persist only for the life of the Communication Driver.	RANGE: -1, 0, 1 -1: Error occurred during saving the configuration file. 0: Always Read value if status is OK. 1: Persist settings (cleared immediately).

Communication Driver Device-Group-Specific System Items

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

System Item Name (Type)	Type/ Access Rights	Description	Values
$SYS$UpdateInterval	Dword/ Read/ Write	Accesses the currently set update interval. It is the current update interval of the device group in milliseconds. A client can poke new values into this item. The value of zero indicates that no non-system items on that topic are updated (data for these items are not acquired from the device).	RANGE: 0…2147483647 0: Topic inactive, no items are updated. Data acquisition is stopped. >0: Expected updated interval for the set of all items in the device group.
$SYS$MaxInterval	Dword/ Read	Accesses the currently measured maximum update interval in milliseconds of all items of the corresponding device group. This item is read-only. The time between two consecutive updates on the item that is updating the slowest is shown.	RANGE: 0…2147483647 0: If update interval is 0 or if the status is false. >0: Measured update interval.

System Item Name (Type)	Type/ Access Rights	Description	Values
$SYS$WriteComplete	Integer/ Read/ Write	Accesses the state of pending write activities on the corresponding device group. On device group creation (adding items to an OPC group) the value of this system item is initially 1, indicating all write activities are complete – no pokes are pending. If values are poked into any items of the device group, the value of this item changes to 0, indicating write activity is currently in progress. If the server completes all write activities, the value of this item changes to 1 if all pokes were successful or to -1 if at least one poke failed. If the value of this item is not zero, the client can poke 1 or -1 to it (poke a 1 to clear errors, or a -1 to test a client reaction on write errors). If the value of this item is zero, it cannot be poked.	RANGE: -1, 0, 1 1: Write complete (no writes are pending – initial state). 0: Writes are pending. -1: Writes completed with errors.

System Item Name (Type)	Type/ Access Rights	Description	Values
$SYS$ReadComplete	Integer/ Read/ Write	Accesses the state of reads on all items in the corresponding device group. The value is unequal 0 if all active items in a device group are read. If at least one item in the device group is activated, this item changes to 0. It changes to 1 if all newly activated items are read successfully or to -1 if at least one item has a non-good quality. Poking a 0 to this item resets the internal-read states of all items in this device group. This resets this item to 0. If all items are read again after this poke, this item changes back to 1 or -1.	RANGE: -1, 0, 1 1: Read complete (all values have been read). 0: Not all values have been read. -1: All values have been read but some have a non-good quality.
$SYS$ItemCount	Dword/ Read	Accesses the number of items in the corresponding device group. This item is read-only.	RANGE: 0…2147483647 >=0: Number of active items.
$SYS$ActiveItemCount	Dword/ Read	Accesses the number of active items in the corresponding device group. This item is read-only.	RANGE: 0…2147483647 >=0: Number of active items.

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

Communication Driver-Specific System Items

The following system items refer to specific information regarding the Communication Driver, the OI Server Manager, and the controllers.

System Item Name	Description	Data Type	Type/ Access Rights
$SYS$RequestSent	Returns the number of message requests originating from the communications module.	VT_UI4	Read-Only
$Sys$ReplyReceived	Returns the number of reply packets received.	VT_UI4	Read-Only
$Sys$UnsolReceived	Returns the number of unsolicited messages received by the communications module.	VT_UI4	Read-Only

System Item Name	Description	Data Type	Type/ Access Rights
$Sys$UnsolReplied	Returns the number of replies sent in response to the unsolicited messages.	VT_UI4	Read-Only
$Sys$RequestErrors	Returns the number of errors for the requests sent.	VT_UI4	Read-Only
$Sys$RequestTimeout	Returns the number of requests that time out.	VT_UI4	Read-Only
$Sys$ResetStatistics	Resets the statistics in the associated hierarchy node.	VT_BOOL	Read-Write
$Sys$TotalPackets Sent	Returns the number of data packets sent.	VT_UI4	Read-Only
$Sys$TotalPackets Received	Returns the number of replies received.	VT_UI4	Read-Only
$Sys$RateSent	Returns the number of packets sent per second.	VT_UI4	Read-Only
$Sys$RateReceived	Returns the number of packets received per second.	VT_UI4	Read-Only
$Sys$ItemUpdateRate	Returns the number of Read items received per second.	VT_UI4	Read-Only
$Sys$ItemWriteRate	Returns the number of Write items sent out per second.	VT_UI4	Read-Only
$Sys$PlcType	Returns the type of the PLC.	VT_BSTR	Read-Only

Communication Driver Redundant Device-Specific System Items

These system items are specific to the Redundant Device.

System Item Name	Type/Access Rights	Description	Values
$SYS$ForceFailover	Boolean/ ReadWrite	This is required to achieve the failover condition to be forced by client. **Note:** By poking a value of "1" (True) into the Force Failover item, a client can conveniently switch to the secondary device.	TRUE, FALSE

System Item Name	Type/Access Rights	Description	Values
$SYS$ActiveDevice	String/Read-Only	This system item will show the current runtime active device.	Node Hierarchy Name
$SYS$FailoverTime	Time/Read- Only	This system item will show the time at which the switch occurred.	Time at which the switch occurred
$SYS$StandbyDevice	String/Read-Only	This system item will show the current runtime standby device.	Node Hierarchy Name
$SYS$SecondaryDeviceStatus	Boolean/Read-Only	This system item will show the status of the secondary device. This is the status of the second device defined in the configuration and is not changed with any failover. RANGE: 0, 1	RANGE: 0, 1 (Contains the value of the system item $SYS$Status)
$SYS$PrimaryDeviceStatus	Boolean/Read-Only	This system item will show the status of the primary device. This is the status of the first device defined in the configuration and is not changed with any failover. RANGE: 0, 1	RANGE: 0, 1 (Contains the value of the system item $SYS$Status)
$SYS$FailoverReason	String/Read-Only	This system item will show the reason for the failover.	Descriptive text "ForceFailover" or the value of the system item $SYS$ErrorText.

Important: The Redundant Hierarchy, including the Device Group, is not hot-configurable, and requires a Reset on the Redundant Hierarchy to effect a configuration change.

Generic OPC Syntax

A Communication Driver is a container for OPC Groups, providing the mechanism for containing and logically organizing OPC items. Within each OPC Group, an OPC-compliant client can register OPC items, which represent connections to data sources in the field device. All access to OPC items is maintained through the OPC Group.

The fully qualified name for an OPC item is the Item ID, equivalent to Item Name. The syntax for specifying a unique Item ID is Communication Driver-dependent. In OPC data acquisition Communication Drivers, the syntax can be as follows:

```
TCPIP.PLC1.R0001
```

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.
- R0001 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 Communication Driver, Access Paths are equivalent to Device Groups. This parameter defines the update interval between the Communication Driver and the field device for accessing the values of data points in the PLC.

Tested GE Hardware

The GESRTP Communication Driver operates only with a standard network interface card in the computer on which it is installed. The GE PLCs must include Ethernet modules or controllers (CPUs) with built-in Ethernet ports.

The following table lists the hardware that has been tested with the GESRTP Communication Driver.

Device	Description	Hardware and Firmware
Series 90-30 PLC	Controller	TCP/IP Ethernet Interface module (catalog number IC693CMM321) OR Controller (CPU) with built-in 10/100Mbs Ethernet port (catalog number IC693CPU364/ IC693CPU372)
Series 90-70 PLC	Controller	MMS-Ethernet Controller module (catalog number IC697CMM741/IC697CMM742) You must properly configure and download device IC697CMM741 using the Series 90-70 Ethernet TCP/IP software (catalog number IC651ENS042).
Versamax	Controller	CPU with Ethernet port (catalog number IC200CPUE05)

Device	Description	Hardware and Firmware
VersamaxMicro PLC VersamaxNano PLC	Controller	Ethernet Interface, a Bridge from RS-232 or RS-485 Serial to Ethernet 10 BaseT (catalog number IC200SET001)
Rx7i PACSystems	Controller	TCP/IP Ethernet Interface module (catalog number IC698ETM001) OR Controller (CPU) with Embedded 10/100Mbs Ethernet port (catalog number IC698CPE010/ IC698CPE020)
Rx3i PACSystems	Controller	TCP/IP Ethernet Interface module (catalog number IC695ETM001) OR Controller (CPU) with Embedded 10/100Mbs Ethernet port (catalog number IC695CPE305)

Chapter 4

Troubleshooting the GESRTP Communication Driver

Troubleshooting

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 GESRTP Communication Driver. Your client, for example, the InTouch software, can also monitor connectivity with the PLC through the $SYS$Status item. Use these tools to troubleshoot your GESRTP Communication Driver.

Finding the Communication Driver Version Number

This section describes finding the version number of your Communication Driver.

To find the version number using Control Panel

1. Use the Windows Control Panel to access the Uninstall or Change a Program window.

   For information on accessing this window in any specific Windows version, refer to your Windows documentation.
2. Locate the Communication Driver in the list of installed programs.
3. Check the **Version** column for the version details of the Communication Driver.

To find the version number using OI Server Manager

- Click on the Communication Driver node in the console tree. The build version numbers of the respective Communication Driver components appear in the details pane.

To find the version number using Windows Explorer

1. Search for **GESRTP.dll**.
2. Right-click on the file name and then click **Properties**. The **Properties** dialog box appears.
3. Click the **Version** tab. The build version number of your Communication Driver is listed under **File Version**.

Diagnostics and Error Tracing

The GESRTP Communication Driver uses the standard diagnostic information provided by the Operations Integration Toolkit. Access to other internal diagnostic registers of the PLC is performed through reads and writes via the syntax used in item naming.

Error Tracing with the Logger

The GESRTP Communication Driver supports error messages, controller-specific error 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 Help for more information about using log flags.

GESRTP Communication Driver Logger Flags

The GESRTP Communication Driver supports the following server-specific OI logger flags.

- Errors
  General errors from the Communication Driver have the prefix "ERROR." They are highlighted in red so that you can find them quickly. Log flags for GESRTP-specific errors have the suffix "_ERROR." They include CONNECTION_ERROR, POLL_ERROR, and POKE_ERROR.
- Trace
  General traces from the Communication Driver have the suffix "_TRACE." The sever-specific trace log flags include CONNECTION_TRACE, POLL_TRACE, and POKE_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.
	CONNECTION_TRACE	Shows connection traces.
	POLL_TRACE	Shows traces of poll messages.

Type	Name	Description
	POKE_TRACE	Shows traces of poke messages.

Error Messages, Codes, and Warnings

In addition to the GESRTP Communication Driver error and warning messages, generic Communication Driver error codes are supported. Use these messages together with the OI Server Manager Diagnostic root data to troubleshoot GESRTP Communication Driver problems.

Note: The logger messages use the following codes: %s to represent strings and %d to represent numbers (integer).

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.

GESRTP Communication Driver Error Messages

The following list contains error messages produced by the Communication Driver that are logged to the Log Viewer with the DASProtFail, DASProtWarn, DASReceiveMessage, and DASSendMessage log flags. These logger messages occur in error situations if the log flag for errors is on.

Logger messages can be useful for debugging communications problems.

Error Message	Explanation	Probable Cause	Solution
DASProtFail Messages			
GESRTP Server creation failed.	The GESRTP Communication Driver cannot create an object.	The Communication Driver is not properly installed or it may be a memory-related issue.	Reinstall the Communication Driver.
Demo license mode for GESRTP Server expired.	The GESRTP Communication Driver was running in demo mode and the time expired.	The demo license time period expired.	Purchase a license.
Failed to initialize listen socket. Hence no unsolicited messages will be received.	The Unsolicited Listen socket failed to initialize.	The socket initialization could not be done properly.	Check the connection and re-start the server.
Attempt to resolve remote hostname "<host_name>" failed.	The hostname could not be resolved.	The hostname specified is not a valid hostname.	Verify the correct IP address of the PLC and the hostname in the hosts file.

Error Message	**Explanation**	**Probable Cause**	**Solution**
Connection attempt for the PLC:<plc_name> has timed out, closing socket at <time>.	The connection attempt to the PLC has timed out.	Network problem.	Check the network connection and try again.
Connection socket to the PLC: <plc_name> failed at <time>.	The socket connection failed.	Network problem.	Check the network connection and try again.
Poke value string length to the Item:'<item_name>' is exceeding 2000 characters. values cannot be processed.	The poke value to the string item is exceeding the protocol limit.	The poke value to the string item is exceeding the protocol limit.	The client must write a smaller value.
<item_name>: Bad Poll Response Received::Error Codes Major:"<major_code>" Minor:"<minor_code>".	The poll operation to an item has a bad response.	The poll could not be done properly.	Check the major and minor code information for more details.
Fault Table: Response packet received from the PLC has insufficient data.	The response packet for the fault number is bad.	The request packet was not sent properly or the fault does not exist in the PLC.	Retry the operation. If the problem persists, contact Technical Support.
Bool items: Response packet from the PLC has insufficient data.	The response packet from the PLC has insufficient data for the advised Boolean items.	The request packet was not sent properly.	Retry the operation. If the problem persists, contact Technical Support.
Insufficient response packet received from the PLC while advising an item.	The response packet from the PLC after advise is bad.	The packet information is lost.	Un-advise and advise again. If the problem persists, contact Technical Support.

Error Message	Explanation	Probable Cause	Solution
Poke operation failed for the item:'<item_name>.	The poke could not be done to the PLC item.	Could not get the poked value from the client.	Retry the poke to the item. If the problem persists, contact Technical Support.
<item_name> : Bad Poke Response Received:: Error Codes Major:<major_code> Minor:<minor_code>	The poke response for the poke operation to an item has a bad response.	The poke could not be done properly.	Check the major and minor code information for more details.
Insufficient response packet received from the PLC while poking to an item.	The response packet from the PLC after the poke is bad.	The packet information is lost.	Retry the poke.
Failed to retrieve host information from a host database. Error code = <error_code>.	The hostname configured for the PLC could not be resolved.	N/A	Use a valid hostname for the PLC.
The socket is marked as nonblocking and the connection cannot be completed immediately for host <host_name>.	A connection could not be established.	The socket could not get the connection established.	Check the network connection and try again.
Socket Read failed with errorcode <error_code>.	The socket read failed while reading a value from the PLC item.	Network problem.	Check the network connection and try again.
SocketWrite failed with errorcode <error_code>.	The socket write failed while updating a value to an item.	Network problem.	Check the network connection and try again.
Unsolicited socket read failed with errorcode <error_code>.	The read from the unsolicited socket failed.	Network problem.	Check the network connection and try again.
Unsolicited socket write failed with errorcode <error_code>.	The write to the unsolicited socket failed.	Network problem.	Check the network connection and try again.
DASProtWarn Messages			

Error Message	Explanation	Probable Cause	Solution
PLC Type configured for the host::%s is not matching with the actual PLC type, this may affect the PLC Type dependent items.	The PLC type selected does not match the actual connected PLC.	Either the wrong PLC type is selected or the specified host name is wrong.	Select the correct setting and re-start the server.
Reply Timeout is hot-configured to <new_value> millisecs for PLC node '<plc_name>'.	The Reply Timeout parameter is changed while server is running.	You changed the parameter.The change takes effect immediately.	N/A
Connection Timeout is hot-configured to <new_value> millisecs for PLC node '<plc_name>.	The Connection Timeout parameter is changed while server is running.	You changed the parameter. The change takes effect immediately.	N/A
Max outstanding messages is hot-configured to <new_value> messages for PLC node '<plc_name>'.	The Max outstanding messages parameter os changed while server is running.	You changed the parameter. The change takes effect immediately.	N/A
Program Name is hot-configured to <new_value> for PLC node '<plc_name>'.	The Program Name parameter is changed while server is running.	You changed the parameter. The change takes effect immediately.	N/A
Value of attribute '<attribute_name>' for PLC node '<plc_name>' has changed - Need to restart the server to make the changed value affective.	The attribute is changed while the server is running and the attribute is not hot configurable.	The value change does not go into effect until the server is restarted.	Restart the server.
'<item_name>' is not a valid item name for this PLC<plc_name>.	The item is not supported by the connected PLC.	The wrong item name was specified.	Specify the correct item name.

Error Message	**Explanation**	**Probable Cause**	**Solution**
Read value of element <register_number> in the Item:'<item_name>'is clamped to BCD value 9999.	The value of the register in the register array is clamped to 9999 BCD value.	The BCD limit was reached.	N/A
ServiceRequest code didn't not match the Code:<service_code>,while updating the data to items.	The service request code did not match the supported items service request code in the response packet.	N/A	If the problem persists, contact Technical Support.
Program Name in the PLC<plc_name> is not matching with the Program Name configured in the PLC faceplate; Hence %P and %L items can't be accessed.	The configured program name does not match the program name running in the PLC.	The wrong program name was specified.	Change (hot configure) the program name.
Mixed Suffix: It is not allowed to mix suffix M with suffix A(Register Array).	An unsupported suffix combination is used.	Suffix M and A are used for a single item. This is not supported.	Un-advise the item.
Mixed Suffix: It is not allowed to mix more than one suffix with suffix A(Register Array).	An unsupported suffix combination is used.	More than one suffix is used with suffix A while advising an item as a register array. This is not supported.	Un-advise the item.
Mixed Suffix: Multiple suffixes to an item are not allowed except with suffix A(Register Array).	An unsupported suffix combination is used.	More than one suffix is used while advising an item and one of the suffixes is not suffix A.	Un-advise the item.
Sub-ProgramName supplied along with %L is not a valid name.	The sub-program name is wrong.	N/A	Check the PLC program and correctly specify the sub-program name.

Error Message	**Explanation**	**Probable Cause**	**Solution**
Item <item_name> cannot be added to the PLC <plc_name> due to invalid range, valid range is between 1 to 65535.	The item could not be advised to the PLC because it is not supported by the PLC.	Either the wrong syntax or an invalid range is specified while advising an item.	Check the valid syntax and ranges of the items supported by the PLC.
Invalid suffix type for ASCII string or Register array type item.	The suffixes specified for the ASCII string or register array items are wrong.	Invalid suffixes are applied for the items.	For a detailed explanation of the items advised with suffixes, see the item reference in this documentation.
Item <item_name> cannot be added to the PLC <plc_name>, because range specified are not registers.	The array registers specified are not valid registers.	The array registers specified are not valid registers.	Advise the item with correct registers.
Item:<item_name> cannot be added, Range specified is more than <packet_size>.	The register array or ASCII item specified is more than the protocol limit data size.	The protocol packet limit is exceeded for the advised item.	Advise the item within the supported limit.
Sub-BlockName for the %L item cannot be more than eight characters.	The wrong sub-block specified is.	The sub-block name specified in the item syntax does not match the sub-block constraints.	Check the sub-block naming constraints in the item reference.
Not a valid bit specification in advising Item as discrete, please specify a valid bit number between 0 and 15.	An invalid bit specification is specified.	An invalid bit specification is specified.	Check the item syntax. The valid bit specification to read is from 0 to 15.
Item <item_name> cannot be added as the Item name is invalid for PLC<plc_name>.	The item specified is not supported by the PLC in advise.	An invalid item is advised to the PLC.	Check the item syntax.

Error Message	**Explanation**	**Probable Cause**	**Solution**
PLC Fault Number is out-of-range, please specify valid PLC fault number between 1 and 16.	The PLC fault table fault number advised is not supported.	The PLC fault table fault number advised is not supported.	Retry with a valid fault number. The supported PLC faults range from 1 to 16.
IO Fault number is out-of-range, please specify the valid IO fault number between 1 and 32.	The IO fault table fault number advised is not supported.	The IO fault table fault number advised is not supported.	Retry with a valid fault number. The supported IO faults range from 1 to 32.
Item <item_name> cannot be added to the PLC <plc_name>, Invalid system item.	An invalid system item is advised.	An invalid system item is advised.	Retry with correct item syntax.
Invalid item name <item_name> for the PLC <plc_name>, Unknown Memory Type.	An unknown item type is advised.	An item type unknown by the server is advised.	Retry with correct item syntax.
Specifying bit position is not valid for boolean items.	A Boolean item is advised for a specific bit position.	It is not allowed to read the bit positions in the Boolean items.	Retry with correct item syntax.
No suffix is allowed with boolean Items.	A Boolean item is advised with a suffix.	Boolean items do not support the item suffixes.	Check the suffix constraints specified in this documentation.
Sub-BlockName for %L items cannot be more than eight characters or NULL.	The specified sub-block is wrong.	The sub-block name specified in the item syntax does not match the sub-block constraints.	Check the sub-block naming constraints in the item reference in this documentation.
Sub-Block Name for %L item provided is more than eight characters.	The specified sub-block is wrong.	The sub-block name specified in the item syntax does not match the sub-block constraints.	Check the sub-block naming constraints in the item reference in this documentation.
Poke value to the item '<register_no>' is clamped to	The Modulo-10000 point output value overflowed. The value is clamped.	The Modulo-10000 value written to the PLC exceeded the maximum limit.	The client must write a smaller value.

Error Message	Explanation	Probable Cause	Solution
<clamped_val>(<actual_val>).			
Poke value to item '<item_name>' is clamped to BCD value 9999(<Poked_value>)	The Modulo-10000 point output value overflowed. The value is clamped.	The Modulo-10000 value written to the PLC exceeded the maximum limit.	The client must write a smaller value.
Poke value to item '<item_name>' is clamped to value <clamped_value>(<poked_value>).	The Modulo-10000 point output value overflowed. The value is clamped.	The Modulo-10000 value written to the PLC exceeded the maximum limit.	The client must write a smaller value.
Poke value to element <register_number> of item '<item_name>' is clamped to value <clamped_value>(<poked_value>).	The Modulo-10000 point output value overflowed. The value is clamped.	The Modulo-10000 value written to the PLC exceeded the maximum limit.	The client must write a smaller value.
Poke value to element <register_number> of item '<item_name>' is clamped to BCD value 9999(<poked_value>).	The Modulo-10000 point output value overflowed. The value is clamped.	The Modulo-10000 value written to the PLC exceeded the maximum limit.	The client must write a smaller value.
Number of values poked to array item '<item_name>' exceeds its size, hence limited to <clamped_value>(<poked_value>).	The values poked to an item exceeded the size of the array item. The value is clamped.	Too many values were poked to an array item.	N/A

Error Message	Explanation	Probable Cause	Solution
Poke value to the item :'<item_name>' is clamped <clamped_val>(<actual_val>).	The value poked to an item exceeds its capacity. The value is clamped.	A value larger than the data type was poked.	Poke a value within the valid range.
%P item is supported by 'Series 9070' PLCs only. Hence item <item_name> cannot be added to PLC<plc_name>	%P item is advised to an unsupported PLC type.	%P and %L items are supported only by Series 90-70 PLCs	Advise the items %P and %L to Series 90-70 PLCs.
%L item is supported by 'Series 9070' PLCs only. Hence item <item_name> cannot be added to PLC<plc_name>	%L item is advised to an unsupported PLC type.	%P and %L items are supported only by Series 90-70 PLCs	Advise the items %P and %L to Series 90-70 PLCs.
DASReceive Messages			
Data socket handling reply '%s	This is a data reply from the Read socket.	N/A	N/A
DASSend Messages			
Unsolicited response send() - operation not complete, wait for event	The socket Write operation for the unsolicited message fails to complete.	The Send operation fails to complete.	Wait for the Send event to complete.
Unsolicited response send() - WSAGetLastError = %d	The socket Write operation for the unsolicited message fails.	The acknowledgement packet for the unsolicited message cannot be sent.	Check the network connection. If the problem persists, contact Technical Support.
Info Messages			
Re-trying connection for the PLC:<plc_name> after slowpoll interval.	The PLC connection failed, so a retry occurs after the specified slow poll interval.	The connection to the PLC failed.	Wait for the connection until the server comes out of slow poll mode.

Communication Driver Error Codes

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

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

GESRTP Communication Driver Error Codes

The following table lists the exception codes that the Communication Driver generates through the SRTP protocol. These error codes, with their server-specific strings, are logged to the Logger. The log flag for all these error codes is DASProtFail.

SRTP Error Code	Logger Message
DASProtFail Messages	
1	Illegal Service Request: either not defined or not supported.
2	Insufficient Privilege: the minor status field contains the privilege level required for the service request.
4	Protocol Sequence Error: the CPU has received a message that is out of order.
5	Service Request Error: Minor status field contains the request specific error code. See table of Minor Error Status Codes below.
6	Illegal Mailbox Type: Service request mailbox type is either undefined or unexpected.

SRTP Error Code	Logger Message
7	The PLC CPU's Service Request Queue is full. The master should retry later. It is recommended that the master wait a minimum of 10 ms before sending another service request.
178	Program block already exists and cannot be replaced.
179	Length limit exceeded; includes read past end of transferred data, writes past end of program block.
180	Attempt to alter interrupt list in MAIN DECL BLOCK during RUN MODE.
181	Additive checksum comparison in Verify failed.
182	Cyclic Redundancy Check (CRC) checksum comparison in Verify failed.
183	Segment length in Verify not equal to the segment length of block in the PLC.
184	Size of the Segment Selector Table in TYPDEF record is not correct.
185	Executable flag in TYPDEF record not set.
186	Block Set already exists, cannot create.
187	Maximum length of a partial store exceeded.
188	Block Type (e.g., data) not found.
189	Block Set (subblock name) not found.
190	Bad Block Type given in Load/Store.
191	Illegal OMF record type/data contents.
192	Bad OMF record checksum in store.
193	Invalid block state transition.
194	The OEM key is NULL (inactive).
195	Text length does not match traffic type.
196	Verify with FA Card or EEPROM failed.
197	No task-level Rack/Slot configuration to read or delete.

SRTP Error Code	Logger Message
198	Control Program (CP) tasks exist but requestor not logged into main CP.
199	Password(s) already enabled and cannot be forced inactive.
200	Password(s) already enabled and cannot be forced inactive.
201	Login using non-zero buffer size required for block commands.
202	Device is write protected.
203	A comm or write verify error occurred during save or restore.
204	Data stored on device has been corrupted and is no longer reliable.
205	Attempt was made to read a device but no data has been stored on it.
206	Specified device has insufficient memory to handle request.
207	Specified device is not available in the system (not present).
208	One or more PLC modules configured have unsupported revision.
209	Packet size or total program size does not match input.
210	Invalid write mode parameter.
211	User Program Module (UPM) read or write exceeded block end.
212	Mismatch of configuration checksum.
213	Invalid block name specified in Datagram.
214	Datagram connection boundary exceeded.
215	Invalid Datagram type specified.
216	Point length not allowed.
217	Transfer type invalid for this selector.
218	Null pointer to data in segment selector.

SRTP Error Code	Logger Message
219	Invalid segment selector in Datagram.
220	Unable to find connection address.
221	Unable to locate given connection ID.
222	Size of Datagram connection invalid.
223	Invalid Datagram connection address.
224	Service in process cannot login.
225	No I/O configuration to read or delete.
226	IOS could not delete configuration or bad type.
227	CPU revision number does not match.
228	Segment for this selector does not exist.
229	DOS file area not formatted.
230	CPU model number does not match.
231	Configuration is not valid.
232	No user memory is available to allocate.
233	Segment selector not valid in context.
234	Not logged in to process service request.
235	Task unable to be deleted.
236	Task unable to be created.
237	VMEbus error encountered.
238	Could not return block sizes.
239	Programmer is already attached.
240	Request only valid in stop mode.
241	Request only valid from programmer.
242	Invalid program cannot log in.
243	I/O configuration mismatch.
244	Invalid input parameter in request.
245	Invalid password.
246	Invalid sweep state to set.

SRTP Error Code	Logger Message
247	Required to log in to a task for service.
248	Invalid task name referenced.
249	Task address out of range.
250	Cannot replace I/O module
251	Cannot clear I/O configuration.
252	I/O configuration is invalid.
253	Unable to perform auto configuration.
254	No privilege for attempted operation.
255	Service request has been aborted.

Data Conversion

The following table describes how the GESRTP Communication Driver 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 smaller than the lower limit, the quality of the item goes to uncertain-LOWLIMITED.

Quality Settings

The GESRTP Communication Driver 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 the data is good.
		The register is read or written to without any problems converting the data.
00C1	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.
00C2	Clamp high	Data communications is good but the data is uncertain.
		The data is clamped at a high limit.
		The register is 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.

Quality Code	Quality State	Description
0004	Bad configure/ No access	This is a bad configuration error.
		Data communications is good but the data cannot be sent and/or received. The data is bad and cannot be used.
		Item cannot be accessed.
		The item does not exist or is not available.
		The server can communicate with the PLC but cannot access the register.
		The server determined the point is not valid.
		The PLC responds that the register does not exist, cannot be read, or cannot be written to.
		The server cannot access a fenced, write-protected, or read-only item.
		The PLC is in a mode that does not permit access to this item.
		The number of data bytes is incorrect but the message is otherwise good.
		The command or op code is invalid but the message is otherwise good.
		The PLC is busy. The server has given up retrying.
0018	No communications	Data communications is down.
		Cannot access the PLC due to a communications error.
		The 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.

Quality Code	Quality State	Description
		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.