

# **DVT OPC Server**

# ***Introduction to DVT OPC Server***

**1**

## **1.1 Introduction to DVT OPC Server**

The DVT OPC Server is an OPC-compliant server that is able to communicate with real-time I/O devices and serve data to OPC clients. The DVT OPC Server was implemented using advanced programming concepts of the current version of the OPC for Process Control (OPC) specification for use in developing next-generation industrial software applications.

The DVT OPC Server communicates with OPC clients and real time DVT SmartImage Sensor devices over TCP/IP. This OPC server is based on the popular Modbus protocol, and can be connected to any SmartImage Sensor compatible I/O hardware.

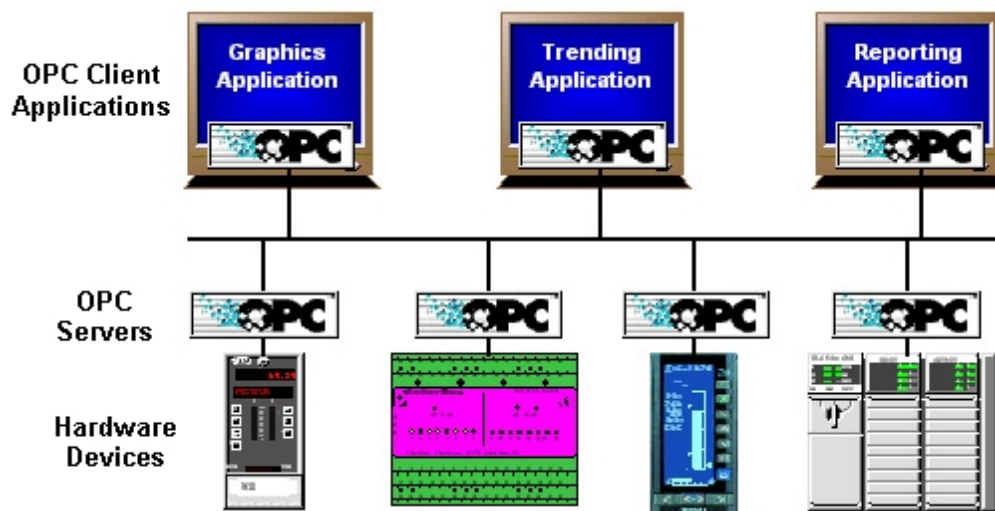
### **1.1.1 Key Features of DVT OPC Server**

The key features of the DVT OPC Server include:

- DVT reserved registers read/write.
- DVT inspection statistics read/write.
- DVT event alarming.
- Advanced OPC data quality and data conversion to client's request.
- Supports multiple SmartImage Sensor devices connected over TCP/IP.
- Internal simulation capabilities for configuration and testing.
- Tag multiplier lets you create hundreds of tags quickly.
- GUI interface for viewing tags, groups, and real-time signals.
- Flexible engineering units and signal ranges.
- Can run as a service or an executable.
- TraceWorX error-logging support.

### 1.1.2 What Is OLE for Process Control?

OLE™ for Process Control (OPC) is a standards-based approach for connecting data sources (e.g., PLCs, controllers, I/O devices, databases, etc.) with HMI client applications (graphics, trending, alarming, etc.). It enhances the interface between client and server applications by providing a universally supported and well-documented mechanism to communicate data from a data source to any client application. Included are not only a detailed guide on how to pass the data, but also specific information on other attributes to supplement those data, such as range information, data type, quality flags, and date and time information. The figure below shows the OPC Architecture, which was introduced by the OPC Foundation. By following the OPC Architecture, a device needs only one standard driver, which is an OPC-compliant server. All OPC-compliant client applications can then be connected to that device, either locally or over a network. Furthermore, connections can be made to more than one OPC server at the same time.



**OPC-Based Client-Server Architecture**

Any OPC client application can connect to any OPC server. In other words, OPC offers true Plug-and-Play capability in the fields of HMI and industrial automation. OPC server types include OPC Data Access (DA), OPC Alarm and Events (AE), and OPC Historical Data Access (HDA).

## 1.2 DVT SmartImage Sensor Supported Protocols

The DVT SmartImage sensors support multiple network communication protocols. The OPC Server must support three of those protocols in order to communicate with the sensors effectively and satisfy the requirements. These three protocols are the:

- Modbus protocol
- Proprietary TCP/IP based DVT Application protocol
- Proprietary TCP/IP based DVT Event Logging protocol

### 1.2.1 Modbus Protocol

The majority of the data transfer between the OPC server and the DVT SmartImage Sensors takes place over the Modbus protocol. The sensors support the Modbus protocol by default through an array of DVT registers, which can be mapped to Modbus registers. The following table lists the **reserved registers** that can be used to monitor and control the functioning of the sensors. The following sections detail the types of data that are transferred using the Modbus protocol.

| DVT Register No. | Bit Position   |             |               |            |            |              |            |             | Type   |
|------------------|----------------|-------------|---------------|------------|------------|--------------|------------|-------------|--------|
|                  | 7              | 6           | 5             | 4          | 3          | 2            | 1          | 0           |        |
| 0                | Output 63      | Output 62   | Output 61     | Output 60  | Output 59  | Output 58    | Output 57  | Output 56   | Output |
| 1                | Output 55      | Output 54   | Output 53     | Output 52  | Output 51  | Output 50    | Output 49  | Output 48   |        |
| 2                | User 16        | User 15     | User 14       | User 13    | User 12    | User 11      | User 10    | User 09     |        |
| 3                | User 08        | User 07     | User 06       | User 05    | User 04    | User 03      | User 02    | User 01     |        |
| 4                | Output 31      | Output 30   | Output 29     | Output 28  | Output 27  | Output 26    | Output 25  | Output 24   |        |
| 5                | Output 23      | Output 22   | Wrong Code    | Strobe 2   | Strobe 3   | Pf Bus Error | Pf Bus Run | CCD Discon. |        |
| 6                | Inspect toggle | Inspecting  | Acquiring     | Run Mode   | Mux 4      | Mux 3        | Mux 2      | Mux 1       |        |
| 7                | Select Fail    | Select Pass | Res. Conflict | Strobe     | Busy       | Fail         | Warn       | Pass        |        |
| 8                | Input 31       | Input 30    | Input 29      | Input 28   | Input 27   | Input 26     | Input 25   | Input 24    | Input  |
| 9                | Input 23       | Input 22    | Input 21      | Input 20   | Input 19   | Input 18     | Release    | Prod Id 14  |        |
| 10               | Prod Id 13     | Prod Id 12  | Prod Id 11    | Prod Id 10 | Prod Id 09 | Prod Id 08   | Prod Id 07 | Prod Id 06  |        |
| 11               | Prod Id 05     | Prod Id 04  | Prod Id 03    | Prod Id 02 | Prod Id 01 | Prod Id 0    | Select     | Trigger     |        |

## DVT Registers

The DVT SmartImage sensors have 16384 registers available. These registers are indexed 0 through 16383. As described in the preceding table, the first 12 registers (0 through 11) are reserved. DVT registers are 8 bits in length (unlike the 16-bit length Modbus registers). Therefore, in order to map a DVT register to a Modbus register the register index number has to be divided by 2. The OPC server is able to read to and write from these registers. Although other protocols may be used to access registers, Modbus is a superior method in that multiple registers can be read and written in one command.

## Digital I/O

The SmartImage Sensors have a number of configurable I/O lines. Legend Series SmartImage Sensors contain eight physical I/O lines, whereas older systems (Series 600) contained twelve physical I/O lines. However, the number of logical I/O lines to which the physical I/O lines can be mapped is much bigger. The number of I/O lines in the Legend Series can be increased to 32 logical inputs and 64 logical outputs by using remote Ethernet I/O devices. All these logical input and output lines can be accessed via the reserved DVT registers, as described in the preceding table. They are marked "Output 0" through "Output 63" for the output lines, and "Input 0" through "Input 31" for the input lines. The OPC server is able to read from and write to these bits (using the Modbus Coils functions).

## Inspection Triggering

The OPC server provides the ability to trigger the sensors to execute inspections from an OPC client. One way to do this is to use the Trigger bit located in the twelfth DVT reserved register. By setting and then clearing the Trigger bit, a toggle effect is simulated and an inspection is triggered. An easier way to trigger an inspection is to use the DVT application protocol as described below.

## Digital Relearn

Digital Relearn is a trigger used by Template, ObjectFind, OCR, Code readers, Spectrograph, and Color SoftSensors to automatically relearn or reset SoftSensor parameters. Relearning occurs when the Digital Relearn input is held active during the next inspection and the SoftSensor has its Digital Relearn parameter checked. The OPC server provides the ability to initiate a Digital Relearn in the sensors. This can be done by setting the Relearn bit located in the 10th DVT reserved register.

## Inspection Product

A SmartImage Sensor may store multiple inspection programs. Each inspection program is called a Product. The currently running product is the Inspection Product.

Every product in a SmartImage Sensor is assigned a number. This number, called the Product Digital ID, is used to reference every product in a SmartImage Sensor. This number enables the user to send a digital signal to the SmartImage Sensor to indicate which product to use for a specific inspection.

The OPC server provides the ability to get and set the inspection product id. One way to set the product id is through the Modbus coil functions: The inspection product id is selected using the reserved DVT registers, as described in the preceding table. The product id bits are marked "Prod Id 0" through "Prod Id 14". An additional bit called "Select" is used to apply the product Id. When the Product Id bits are set, the Select bit is toggled to select the new product.

### 1.2.2 DVT Application Protocol

All DVT SmartImage Sensors implement the DVT Application Protocol (or DVT System Driver). This protocol provides a rich set of functions to manage the sensors. It is largely used by the Framework user interface over the TCP/IP protocol (on port 3246). The OPC server uses the DVT Application Protocol to make remote function calls to the sensor to obtain inspection statistics and to configure event logging.

### 1.2.3 DVT Event Logging Protocol

The SmartImage Sensors have the ability to log events to an Event Logging service on the TCP/IP network. When the IP address and a port number of an event-logging service are specified in the SmartImage Sensor, the event logging can be activated. When a SmartImage Sensor activates event logging, it creates a TCP/IP connection to the service and can start logging events. The OPC server acts as an Event Logging service in order to receive these events and expose them to OPC clients. The OPC Server does not log the events locally, but rather uses the OPC Alarms and Events mechanism to notify the OPC clients.

The default port number for event logging is port 50000. However, this value is configurable through the OPC Server configuration user interface.

## 1.3 Installing DVT OPC Server

This section describes the steps for installing the DVT OPC Server. Be sure to close any other applications before installing DVT OPC Server.

### 1.3.1 System Requirements

To use DVT OPC Server, you must have the following *minimum* system requirements:

- 233 MHz processor
- 64 MB RAM
- 100 MB disk space
- 8X speed CD-ROM drive
- VGA video card (256 or more colors for best results)
- Microsoft® Windows® 2000 with the latest service packs, Windows XP with the latest service packs, Windows NT® 4.0 with the latest service packs, or Windows Embedded NT and XP.

Performance will improve with higher specification hardware (faster data updates).

### 1.3.2 Installation Procedure

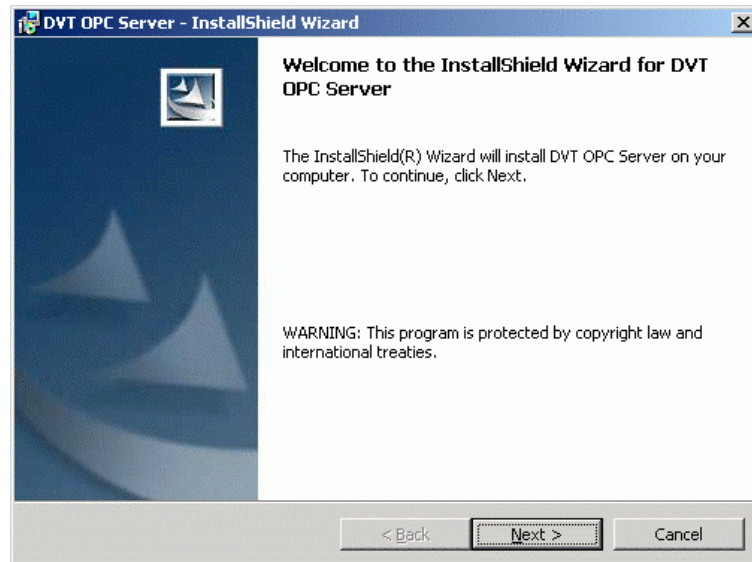
The following steps detail the DVT OPC Server installation:

**Note:** If your operating system (e.g., Windows NT) requires a login name, you must log in with administrator capability before installing DVT OPC Server.

1. Before installing the software, be sure that all other applications are closed and/or disabled.

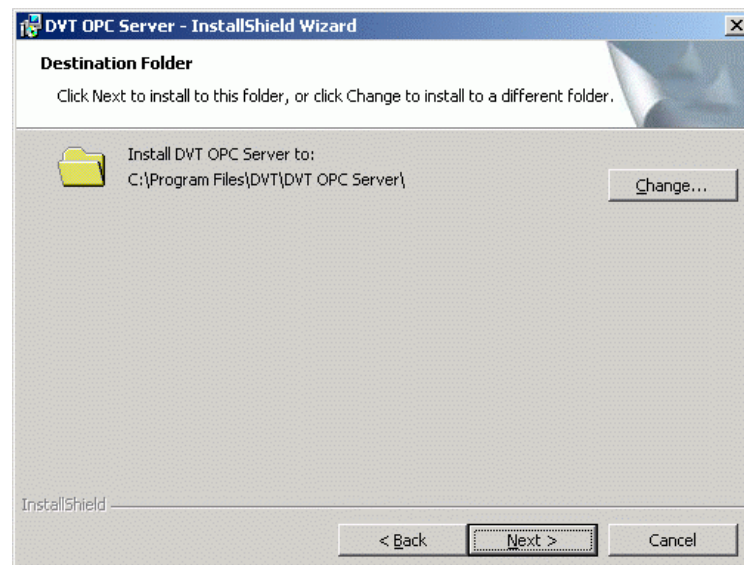
2. Insert the product CD into your CD-ROM drive. The **Welcome** screen appears, as shown in the figure below. Click the **Next** button to continue.

**Note:** When installing the OPC server from an FTP download, begin by locating and launching the **setup.exe** file.



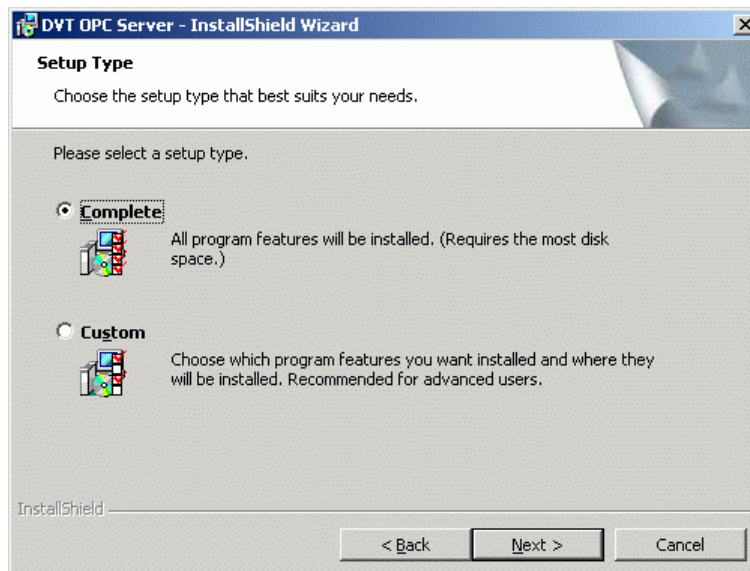
**Welcome Screen**

3. The **Destination Folder** dialog box appears, as shown in the figure below. Choose the destination location for the software installation. If you wish to keep the default location, click the **Next** button to continue. If you prefer to select a different directory location for the installation, click the **Change** button to select a different drive or directory. Click **Next** to continue.



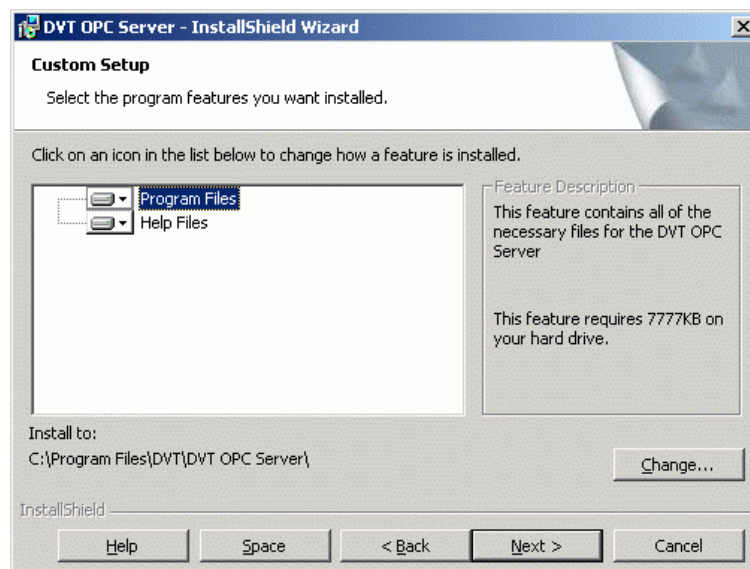
**Choosing the Installation Folder**

4. The **Setup Type** dialog box appears, as shown in the figure below. To install all product components and features, select **Complete**. To choose specific product features to install, select **Custom**. Click **Next** to continue.



#### Ready to Install the Program

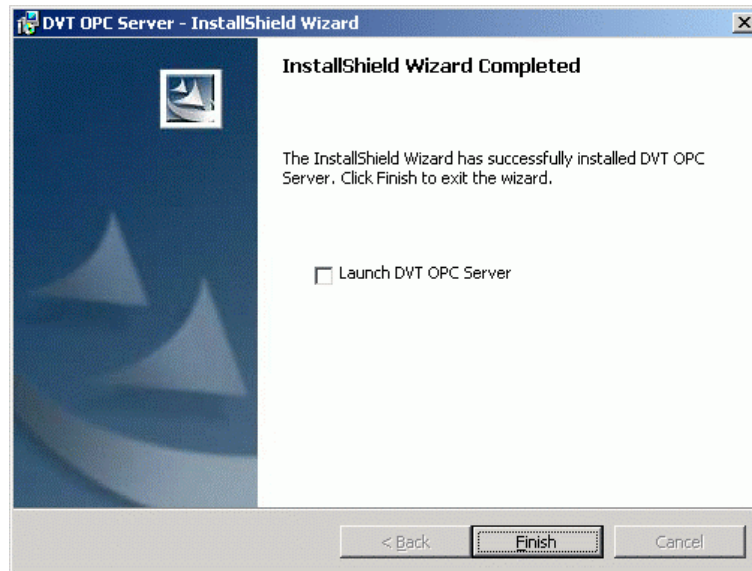
5. In the **Custom Setup** dialog box, shown in the figure below, you can select choose which product components and features to install. Click **Next** to continue with the product installation.



#### Installation Complete

6. After all files and components have been installed, the **Setup Complete** dialog box appears, as shown in the figure below. Click the **Finish** button to complete the installation.

**Note:** It is recommended that you restart your computer before using the product.



**Installation Complete**

## 1.4 DVT OPC Configuration Quick Start

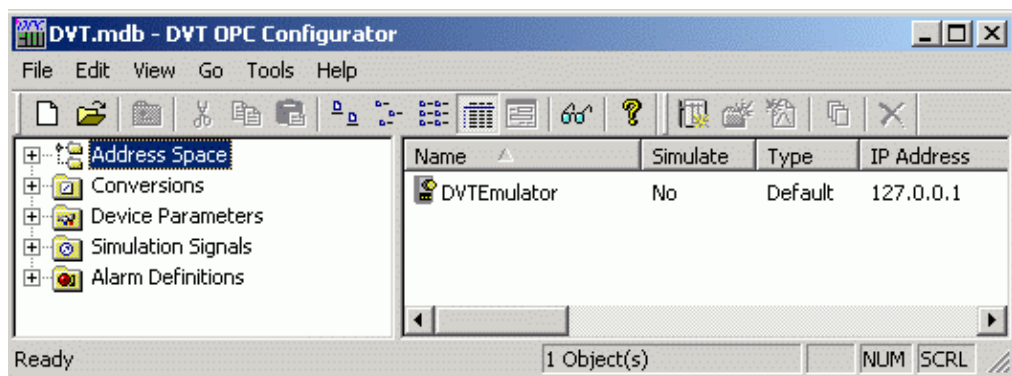
The DVT Configurator is the interface that you use to configure OPC data items. Configuring the DVT OPC server requires the following basic steps:

1. Start the DVT OPC Configurator.
2. Create a new configuration database.
3. Create a new device.
4. Create and configure data items.
5. Activate the configuration database.
6. Connect to the server using an OPC client.

### 1.4.1 Creating a New Configuration Database

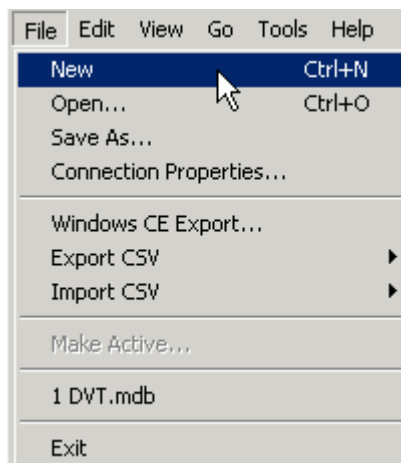
You must first create a database in which your configuration can be stored:

1. From the Windows **Start** menu, select **Programs > DVT Applications > DVT OPC Server > DVT OPC Configurator**.
2. This opens the Configurator, as shown in the figure below. The screen consists of a split window with a tree control view in the left-hand pane and a configuration view in the right-hand pane. The Configurator provides a standard format for the configuration database, as well as a sample (default) configuration project. The Configurator also includes toolbars and menus with many command functions.



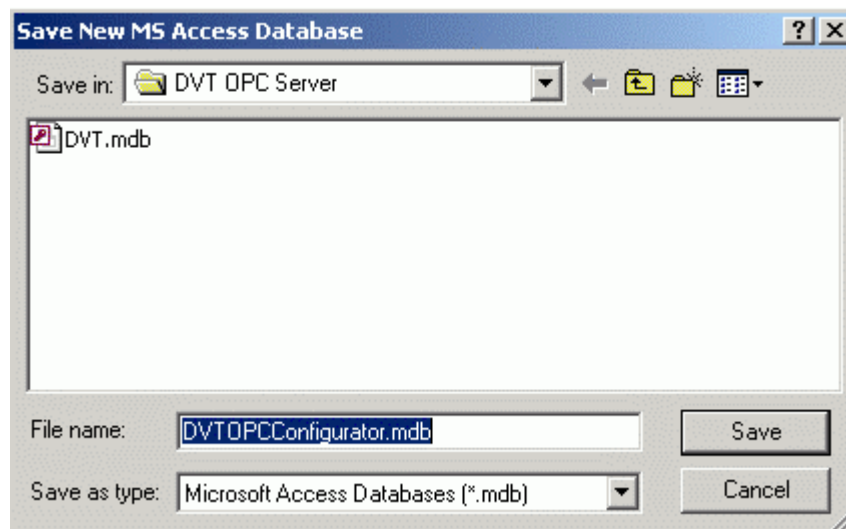
**Configurator Screen**

3. The Configurator uses Microsoft Access configuration databases. To create a new configuration database in the Configurator: Select **New** from the **File** menu, as shown in the figure below.



### Creating a Configuration Database

4. In the **Save New Database** dialog box, select the database type from the **Save As Type** drop-down list, as shown in the figure below. Browse for the target directory, give the file a name, and then click the **Save** button.

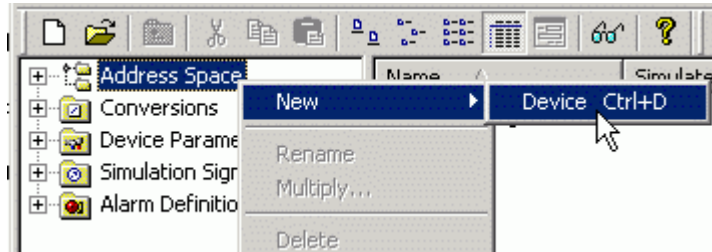


### Saving the New Configuration Database

### 1.4.2 Creating a New Device

To create a new device:

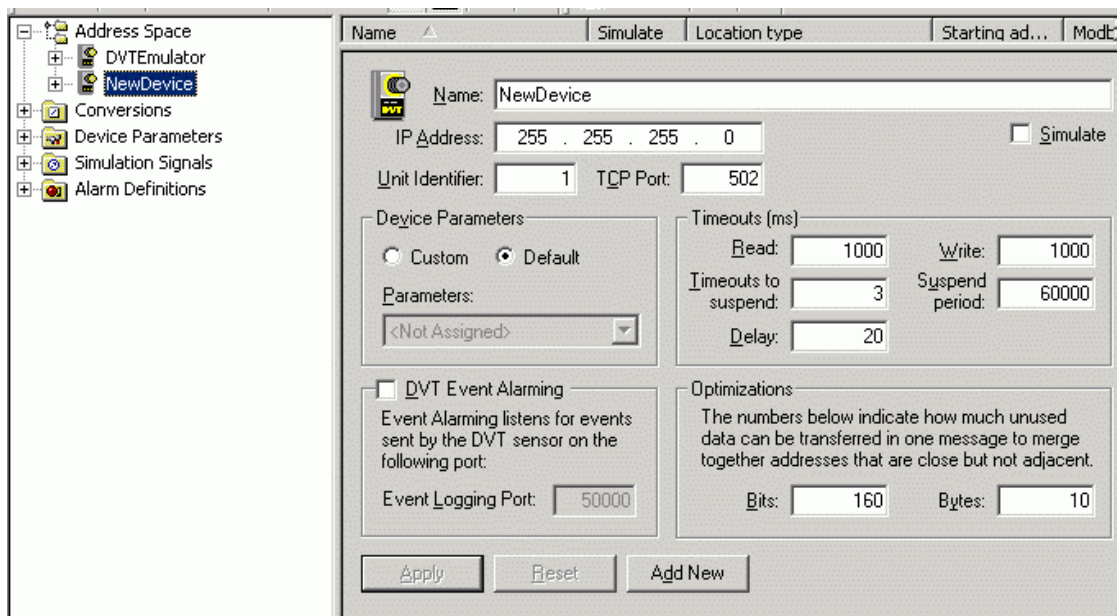
1. Right-click the **Address Space** tree control of the Configurator screen and select **New > Device** from the pop-up menu, as shown in the figure below.



#### Creating a New Device

2. This opens the properties dialog box for the device, as shown in the figure below. In the name field, specify a name for the new device.
3. Configure the properties for the new device. Click the **Apply** button to save your changes.
4. The new device appears under the **Address Space** tree control. Folders containing the inspection statistics and the reserved registers for this device will automatically be created and listed under it.

**Note:** Tags (data items) in the **InspectionParameters** folder, the **InspectionStatistics** folder, and the **ReservedRegistersBits** folder cannot be cut, pasted, copied, or deleted.

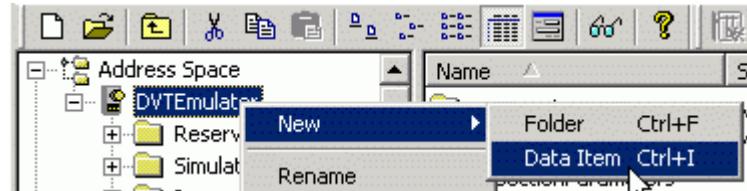


#### Setting the Properties for the New Device

### 1.4.3 Creating a New Data Item

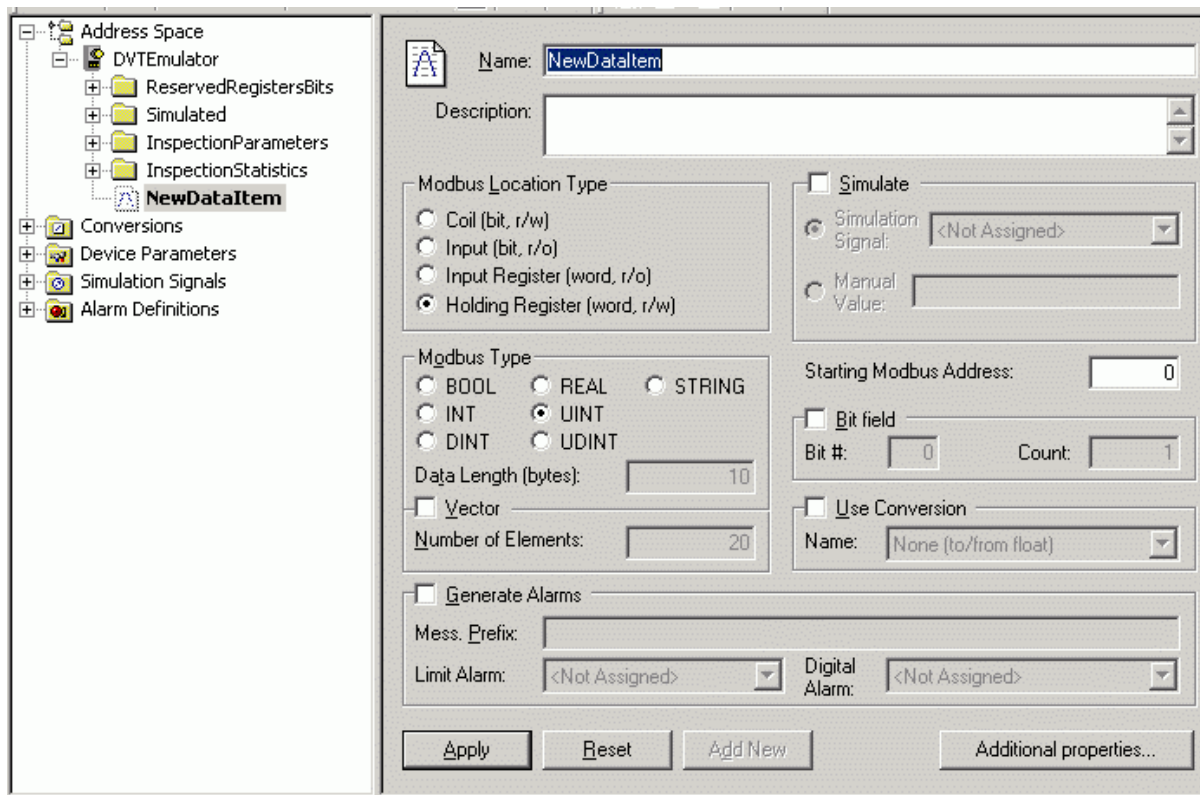
To create a new data item:

1. Right-click a device or folder on the tree control of the Configurator screen and select **New > Data Item** from the pop-up menu, as shown in the figure below.



#### Creating a New Data Item

2. This opens the properties dialog box for the data item, as shown in the figure below. In the name field, specify a name and a description for the new data item.
3. Configure the properties for the new data item. Click the **Apply** button to save your changes.
4. The new data item appears under the tree control.



#### Setting the Properties for the New Data Item

#### 1.4.4 Activating the Database

Once your configuration is complete, you need to make sure that it is the active database. The database that is currently active is the one that the server uses. To make the current database active, select **Make Active** from the **File** menu. A dialog box appears showing both the current active database and database that is currently being edited, as shown in the figure below. To set the edited database as the active database, click the **Yes** button. Next time the server goes into runtime, it will use this active database for all of its operations.



**Activating the Database**

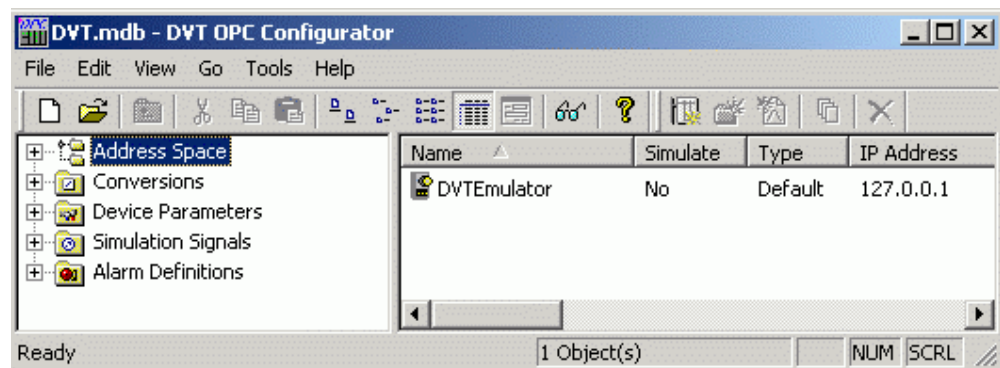
# Starting DVT OPC Server Configuration 2

## 2.1 Starting the Configurator

The OPC server Configurator module is used to configure data items (tags). This configuration is stored to the configuration database (Microsoft Access database file \*.mdb). The OPC server runtime module reads this database on startup and creates its **Address Space** according to the database.

To start the Configurator:

1. From the Windows **Start** menu, select **Programs > DVT Applications > DVT OPC Server > DVT OPC Configurator**.
2. This opens the Configurator, as shown in the figure below. The screen consists of a split window with a tree control view in the left-hand pane and a configuration view in the right-hand pane. The Configurator provides a standard format for the configuration database, as well as a sample (default) configuration project. The Configurator also includes toolbars and menus with many command functions.

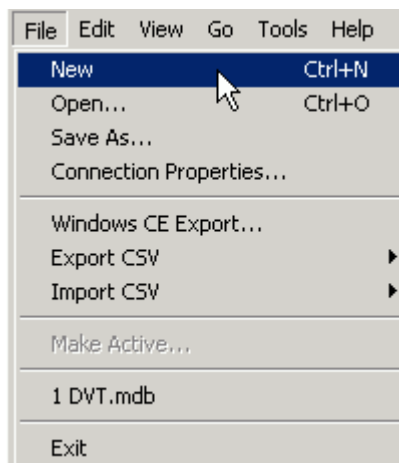


Configurator Screen

## 2.2 Creating Configuration Databases

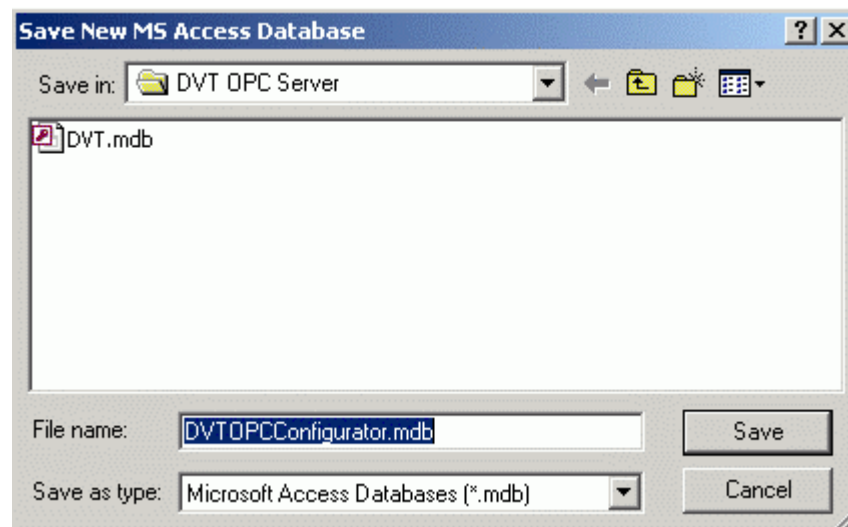
The Configurator uses Microsoft Access configuration databases. To create a new configuration database in the Configurator:

1. Select **New** from the **File** menu, as shown in the figure below.



### Creating a Configuration Database

2. In the **Save New Database** dialog box, select the database type from the **Save As Type** drop-down list, as shown in the figure below. Browse for the target directory, give the file a name, and then click the **Save** button.



### Saving the New Configuration Database

## 2.3 Toolbars

The Configurator contains two toolbars: a **Standard** toolbar and a **Data Manipulation** toolbar.

### 2.3.1 Standard Toolbar

To show or hide the **Standard** toolbar, select **Toolbars > Standard Buttons** from the **View** menu. The **Standard** toolbar, shown below, contains the following command buttons.

|                                                                                   |                                                                     |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------|
|  | <b>New:</b> Creates a new configuration database.                   |
|  | <b>Open:</b> Opens an existing configuration database.              |
|  | <b>Up One Level:</b> Moves up one level in the tree control.        |
|  | <b>Cut:</b> Deletes current selection, sending it to the clipboard. |
|  | <b>Copy:</b> Copies the current selection to the clipboard.         |
|  | <b>Paste:</b> Pastes the current contents of the clipboard.         |
|  | <b>Large Icons:</b> Displays items as large icons.                  |
|  | <b>Small Icons:</b> Displays items as small icons.                  |
|  | <b>List:</b> Displays items as a list.                              |
|  | <b>Details:</b> Displays items as a list with details.              |
|  | <b>Dialog View:</b> Displays additional configuration options.      |
|  | <b>Monitor View:</b> Displays OPC server data in a separate pane.   |
|  | <b>About:</b> Displays information about the application.           |

### 2.3.2 Data Manipulation Toolbar

To show or hide the **Data Manipulation** toolbar, select **Toolbars > Data Manipulation Buttons** from the **View** menu. The **Data Manipulation** toolbar, shown below, contains the following command buttons.

|                                                                                     |                                                                       |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
|  | <b>New Device:</b> Inserts a new device into the configuration.       |
|  | <b>New Folder:</b> Inserts a new folder into the configuration.       |
|  | <b>New Data Item:</b> Inserts a new data item into the configuration. |
|  | <b>Multiply:</b> Duplicates the selected item in the tree control.    |
|  | <b>Delete:</b> Deletes the selected item in the tree control.         |

## 2.4 Menus

The menu bar of the Configurator contains the following menus:

- File
- Edit
- View
- Go
- Tools
- Help

**Note:** You can also access many of the menu commands by right-clicking items in the tree control of the Configurator and selecting command functions from the pop-up menus.

### 2.4.1 File Menu

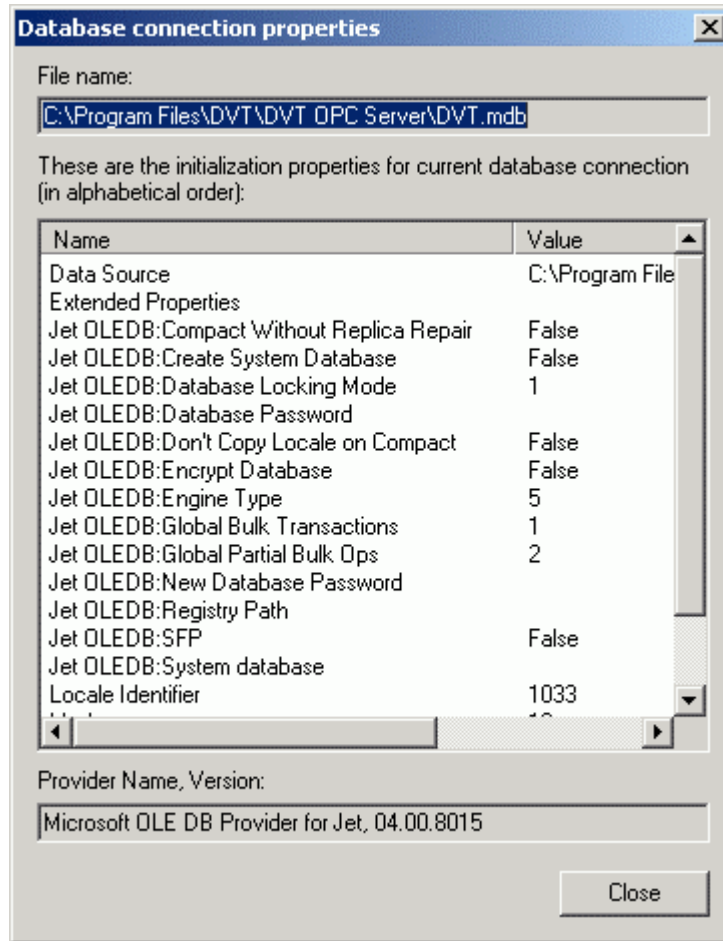
The **File** menu commands are listed in the table below.

**File Menu Commands**

| Command               | Shortcut Keys | Function                                                                                                                                                                                                                                             |
|-----------------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| New                   | CTRL+N        | Creates a new configuration database.                                                                                                                                                                                                                |
| Open                  | CTRL+O        | Opens an existing configuration database.                                                                                                                                                                                                            |
| Save As               |               | Saves the current database under a different name.                                                                                                                                                                                                   |
| Connection Properties |               | Displays the current database connection properties.                                                                                                                                                                                                 |
| Windows CE Export     |               | Exports all data into an extended .csv file. This file can then be transferred to a Windows CE device and used by the server running there. This additional export step is necessary, as Windows CE does not support .mdb or other database formats. |
| Export CSV            |               | Exports configuration data from your database to a text file (.txt) or a Microsoft Excel file (.csv). You can specify the delimiters and what to export.                                                                                             |
| Import CSV            |               | Imports data into your configuration database from a text file (.txt) or a Microsoft Excel file (.csv). You can then specify the delimiters and choose from the import settings.                                                                     |
| Make Active           |               | Makes the current database active. If this command is not available, then the current database is already the active database.                                                                                                                       |
| Exit                  |               | Closes the application. The current configuration database is automatically saved.                                                                                                                                                                   |

## 2.4.2 Database Connection Properties

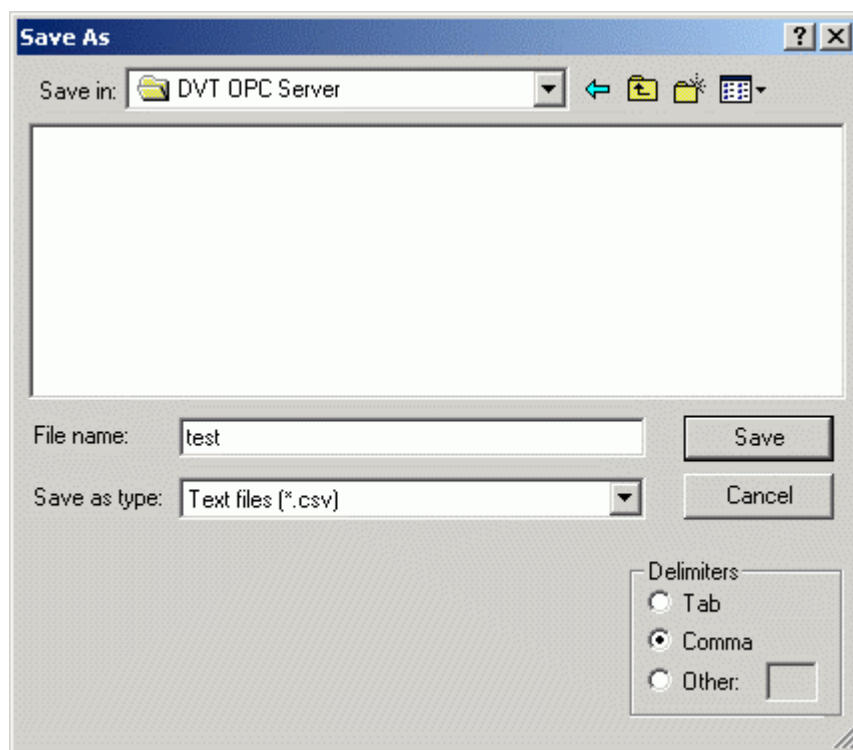
Selecting **Connection Properties** from the **File** menu opens the **Database Connection Properties** dialog box, shown below, which lists the initialization properties for the current database connection.



Database Connection Properties Dialog Box

## 2.4.3 Exporting Configuration Data

The Configurator offers the flexibility of exporting data from your configuration database to a text (.txt) file or a Microsoft Excel (.csv) file. To export data, select **Export CSV** from the **File** menu. You can export all data, or you can choose to export only certain kinds of data. This opens the **Export Configuration Data to File** dialog box, as shown in the figure below. You can then specify the delimiters for exporting the data. Unless you specify delimiters in the **Export Configuration Data to File** dialog box, the file uses **Commas** as delimiters by default. Each group contains headings and columns that provide information about each item, such as descriptions and associated translations and expressions. It also provides the "tree" pathway for each item. Choose the directory to which you want to export the data from your database. In the **Save As Type** field, choose the file type (.txt or .csv) that you would like to save.



Exporting Configuration Data

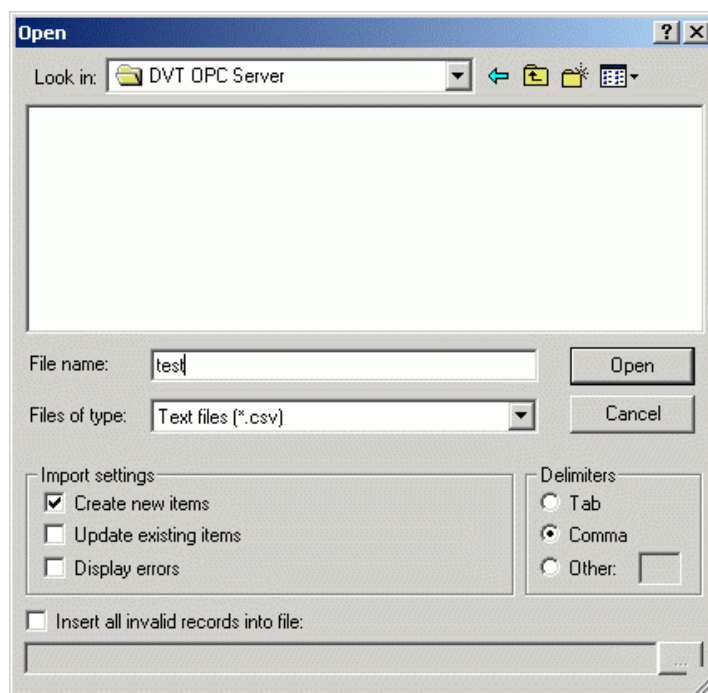
#### 2.4.4 Importing Configuration Data

The Configurator offers the flexibility of importing data from a text (.txt) file or a Microsoft Excel (.csv) file to your configuration database. To import data, select **CSV Import** from the **File** menu. You can import all data, or you can choose to import only certain kinds of data. This opens the **Import Configuration Data From File** dialog box, shown below. You can then specify the delimiters and choose from the following import settings:

- **Create new items:** When the import file contains items that are not yet in the configuration database, then it creates them. Otherwise it skips these items.
- **Update existing items:** When the import file contains items that are in the configuration database, then it updates them using data from the import file. Otherwise it skips these items.

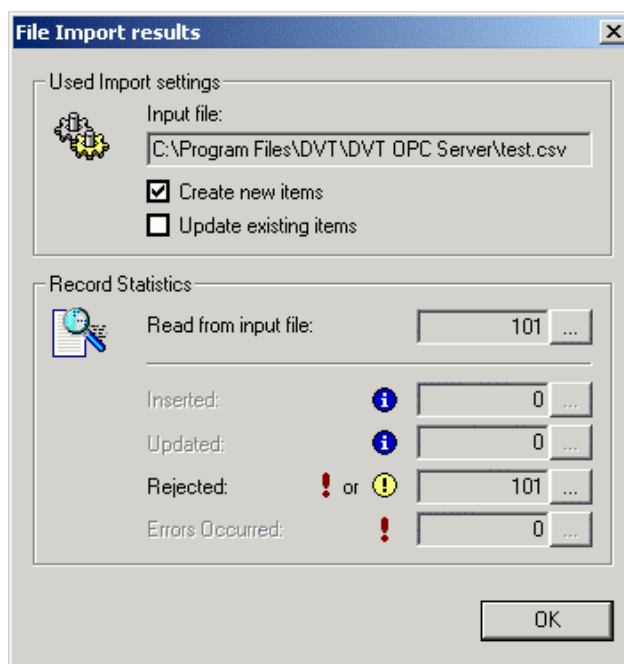
**Note:** Either **Create new items** or **Update existing items** must be selected. Otherwise there is nothing to import.

- **Display errors.** When this item is checked, the Configurator shows a dialog box if an error occurs, and then asks you if you want to proceed with the import. When it is not checked, it skips all items where an error occurred.
- **Insert Missing Parent Items:** When the import file contains parent items that are not yet in the configuration database, then it creates them within the tree control structure of the database. This item is checked by default.



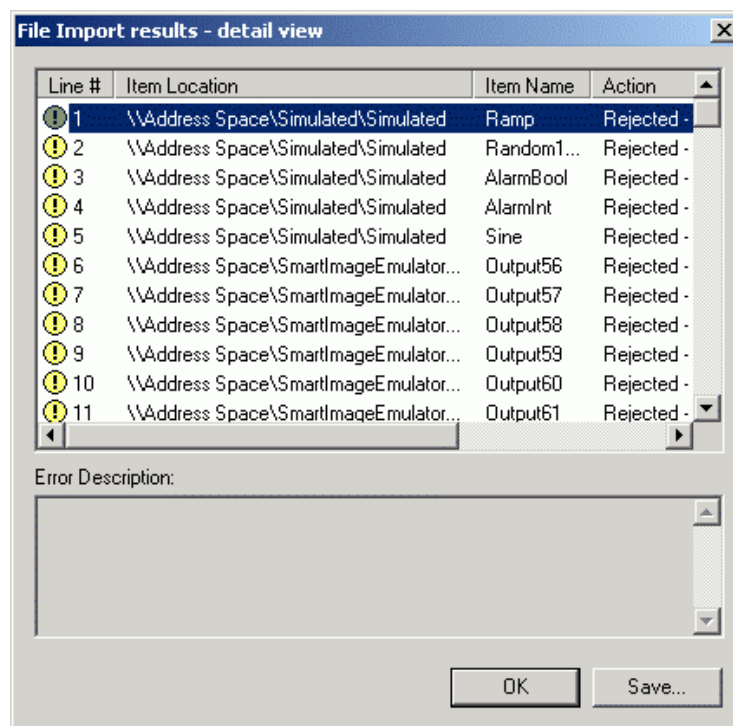
### Importing Configuration Data

When you have selected a file to import, click **Open**. When the import is completed, the **File Import Results** dialog box opens, as shown below. This shows the import settings, including the input file name. It also provides a summary of the import, including how many items were inserted, updated, or rejected, and shows how many errors occurred.



### File Import Results Dialog Box

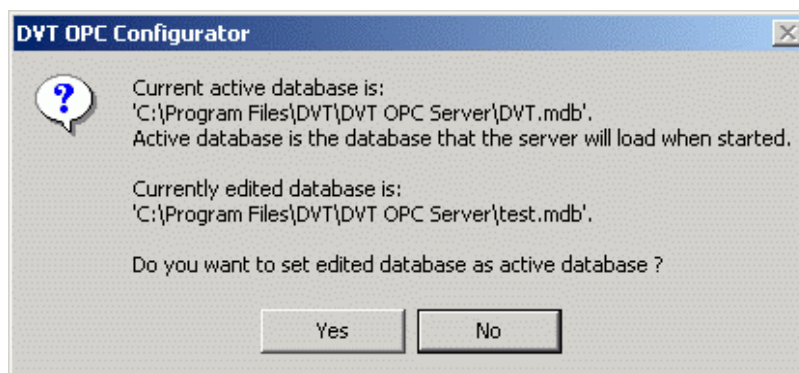
Click the ... button to the right of each field to get the details view of the import results, as shown in the figure below. This view shows the specific items that were inserted, updated, or rejected, as well as a description of any errors that occurred.



**File Import Results Details**

### 2.4.5 Activating the Database

Once your configuration is complete, you need to make sure that it is the active database. The database that is currently active is the one that the server uses. To make the current database active, select **Make Active** from the **File** menu. A dialog box appears showing both the current active database and database that is currently being edited, as shown in the figure below. To set the edited database as the active database, click the **Yes** button. Next time the server goes into runtime, it will use this active database for all of its operations.



**Activating the Database**

## 2.4.6 Edit Menu

The **Edit** menu commands are listed in the table below.

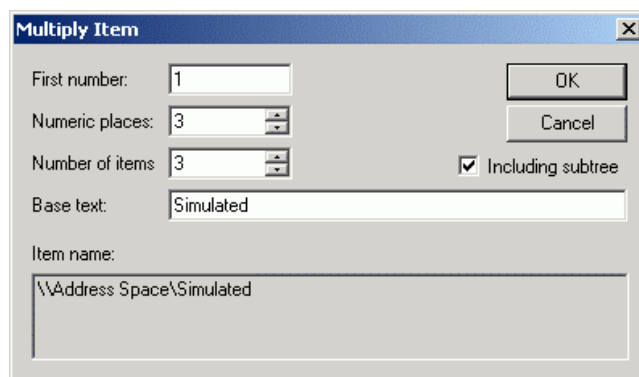
### Edit Menu Commands

| Command          | Shortcut Keys | Function                                                                                                        |
|------------------|---------------|-----------------------------------------------------------------------------------------------------------------|
| New              |               | Creates a new item depending on what is selected in the tree control.                                           |
| Rename           | CTRL+R        | Renames the selected item.                                                                                      |
| Multiply         | CTRL+M        | Opens the <b>Multiply Item</b> dialog box, which allows you to duplicate the selected item in the tree control. |
| Delete           | CTRL+DEL      | Deletes the selected object.                                                                                    |
| Cut              | CTRL+X        | Cuts the selected object from the view and places it on the clipboard.                                          |
| Copy             | CTRL+C        | Copies the selected object to the clipboard.                                                                    |
| Paste            | CTRL+V        | Pastes the last object placed on the clipboard.                                                                 |
| Select All       | CTRL+A        | Selects all objects in a list. The selection is shown in the upper-right-hand section of the viewer.            |
| Invert Selection |               | Unselects all selected items and selects all unselected items.                                                  |

## 2.4.7 Multiplying Items

The Configurator allows you to multiply items in the tree control. Multiplication provides a simple way of developing configurations where there are many similar items in a given category. To multiply an item:

1. Select the item in the tree control that you wish to multiply.



**Multiply Item Dialog Box**

2. Either right-click the item and select **Multiply** from the pop-up menu, or select **Multiply** from the **Edit** menu. This opens the **Multiply Item** dialog box, shown above.
3. When the items are multiplied, they are all given a base name followed by a number. The default base text is the name of the item selected for multiplication. To modify the base text, change the **Base Text** field appropriately.
4. In the **First Number** field, specify the number to appear next to the first multiplied item.
5. In the **Number of Items** field, specify how many items you wish to create.
6. In the **Numeric Places** field, specify the minimum length of each number to append. Values that take up less space than the specified amount of numeric places will have zeros before the number.
7. If you want to multiply all subfolders as well, check the **Including Subtree** check box.
8. Click the **OK** button to do the multiplication. The example configuration shown in the **Multiply Item** dialog box above creates three new items with the following names:
  - Simulated001
  - Simulated002
  - Simulated003

## 2.4.8 View Menu

The **View** menu commands are listed in the table below.

**View Menu Commands**

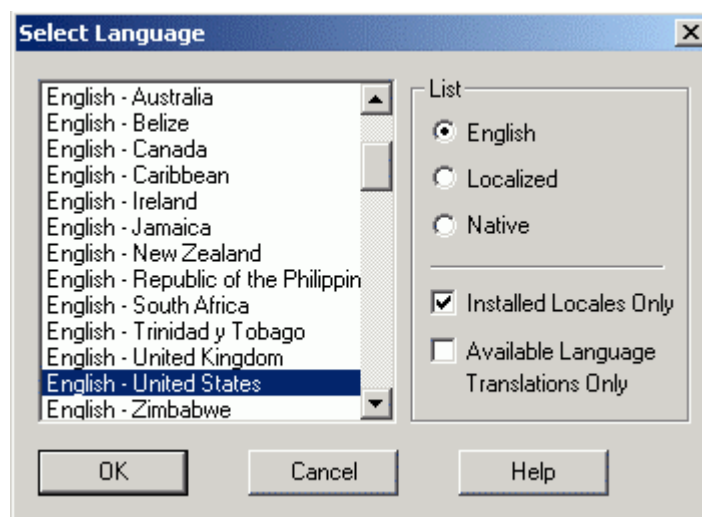
| Command     | Shortcut Keys | Function                                                                                       |
|-------------|---------------|------------------------------------------------------------------------------------------------|
| Toolbars    |               | Shows/hides the Configurator toolbar buttons.                                                  |
| Status Bar  |               | Toggles the status bar at the bottom of the Configurator window.                               |
| Large Icons | F7            | Displays items as large icons.                                                                 |
| Small Icons | F8            | Displays items as small icons.                                                                 |
| List        | F9            | Displays items as a list.                                                                      |
| Details     | F10           | Displays items as a list along with detailed information about the configuration of each item. |
| Dialog View | F11           | Toggles the configuration window (right-hand pane).                                            |

|                   |         |                                                                                                                                                                                                             |
|-------------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Monitor View      | F12     | Displays OPC server data in a separate pane.                                                                                                                                                                |
| Sort By           |         | Displays a list of options for sorting the columns in the right-hand pane of the screen. The options listed depend on the level within the view.                                                            |
| Show/Hide Columns |         | Displays a list of options that you can choose to show or hide in the view.                                                                                                                                 |
| Select Language   |         | Opens the <b>Select Language</b> dialog box. Choose the language you wish to use for your system and click <b>OK</b> . For navigation purposes, use the buttons and check boxes in the <b>List</b> section. |
| Global Refresh    | F5      | Refreshes the data for the entire Configurator screen.                                                                                                                                                      |
| Subtree Refresh   | CTRL+F5 | Refreshes only the data contained in the currently selected subtree.                                                                                                                                        |

## 2.4.9 Selecting Languages

The **Select Language** function on the **View** menu allows you to choose which language to use in your display. Choosing **Select Language** from the **View** menu opens the **Select Language** dialog box, shown in the figure below.

**Note:** A language resource .dll is required for language switching.



Select Language Dialog Box

Define the parameters listed in the table below. Then click **OK** to return to the work area.

**Select Language Parameters**

| Parameter                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| List                                 | Lists available languages. Depending on which item you have selected, the view on the left will change. If <b>English</b> is checked, the languages will appear as their English name. If <b>Localized</b> is checked, the languages will appear with the native country in parentheses (for languages with several dialects only). When <b>Native</b> is checked, the languages are displayed the way they would be written in that language. |
| Installed Locales Only               | If this is checked, local languages appear in the box.                                                                                                                                                                                                                                                                                                                                                                                         |
| Available Language Translations Only | Checking this box allows you to choose from available language translations only.                                                                                                                                                                                                                                                                                                                                                              |

**2.4.10 Go Menu**

The **Go** menu commands are listed in the table below.

**Go Menu Commands**

| Command       | Shortcut Keys   | Function                                                           |
|---------------|-----------------|--------------------------------------------------------------------|
| Up One Level  |                 | Moves the cursor up one level in the tree control.                 |
| Next Item     | ALT+Down Arrow  | Moves the cursor to the next visible item in the tree control.     |
| Previous Item | ALT+Up Arrow    | Moves the cursor to the previous visible item in the tree control. |
| Expand Item   | ALT+Left Arrow  | Expands an item that contains a submenu.                           |
| Collapse Item | ALT+Right Arrow | Collapses an item that contains a submenu.                         |
| Page Up       | ALT+PgUp        | Moves the cursor up to the first item in the tree.                 |
| Page Down     | ALT+PgDown      | Moves the cursor down to the last visible item in the tree.        |
| Home          | ALT+Home        | Moves the cursor up to the first item in the tree.                 |
| End           | ALT+End         | Moves the cursor down to the last visible item in the tree.        |
| Next Pane     | F6              | Moves the cursor to the next pane in the window.                   |
| Previous Pane | SHIFT+F6        | Moves the cursor to the last pane used.                            |

### 2.4.11 Tools Menu

The **Tools** menu launches the **Options** dialog box, which is described in the "Options" section below.

### 2.4.12 Help Menu

The **Help** menu commands are listed in the table below.

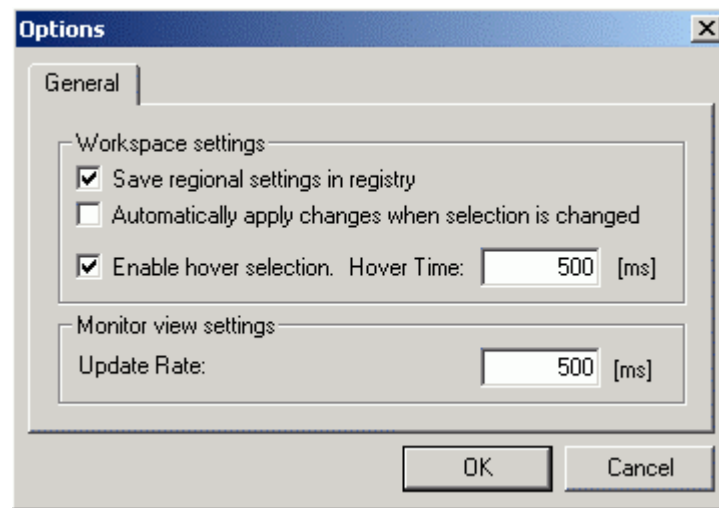
#### Help Menu Commands

| Command           | Shortcut Keys | Function                                                                                                   |
|-------------------|---------------|------------------------------------------------------------------------------------------------------------|
| Help Topics       | F1            | Launches the online Help for the Configurator.                                                             |
| About Application |               | Launches the <b>About Box</b> , which contains information about the product version number and copyright. |

## 2.5 Options

Selecting **Options** from the **Tools** menu opens the **Options** dialog box. The **General** tab of the **Options** dialog box, shown below, contains the following options:

- **Save regional settings in registry:** Checking this option allows you to save regional settings in the registry so that they are applied each time you start the configuration application. This applies to the language settings as well as time and date settings.
- **Automatically apply changes when selection is changed:** Checking this option allows changes to the configuration database to be saved each time you switch dialogs without clicking on the **Apply** button or being shown a message asking if you would like to apply changes.
- **Enable hover selection:** Checking this option allows you to highlight an item by moving the mouse pointer over that item and keeping it there for a specified amount of time (in milliseconds).
- **OPC Monitoring Update Rate:** This specifies the update frequency (in milliseconds) of the items in the Monitor View. The monitor scans the server and displays the tag values at the bottom of the Configurator screen.

**Options Dialog Box: General Tab**

# Address Space

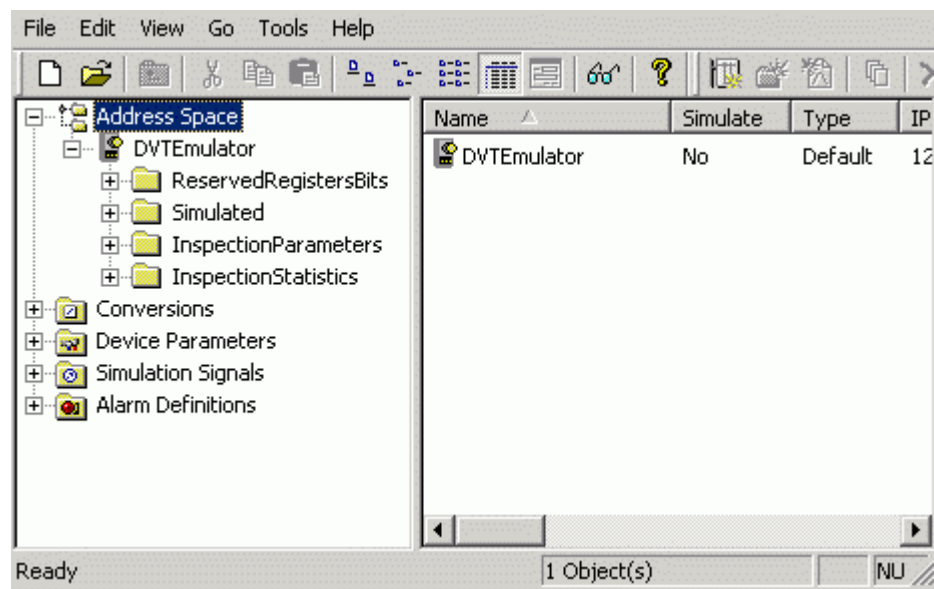
3

## 3.1 Configuring the Address Space

**Modbus/TCP** is a variant of the Modbus family of simple, vendor-neutral communication protocols intended for supervision and control of automation equipment. Specifically, it covers the use of Modbus messaging in an intranet or Internet environment using the TCP/IP protocols. The most common use of the protocols at this time is for Ethernet attachment of PLCs, I/O modules, and "gateways" to other simple field buses or I/O networks.

The **Address Space** tree control in the Configurator, shown in the figure below, sets the properties and connection parameters for the following items:

- Devices
- Folders
- Data Items



Address Space Tree Control

Every device is identified by its unique address. Its registers are read as Input (1 bit long) or Input Register (16 bits), or written to as Coil (1 bit) or Holding Register (16 bits). Registers of each type are addressed by using 16-bit numbers.

Data items can be organized hierarchically. It is similar to organizing directories and files on your computer's hard disk. The OPC Server offers several levels of hierarchy. The Configurator module uses the terms *Folder* and *Data Item*. A **folder** can contain additional folders and also data items. The **data items** are always the branches in the tree control hierarchy.

The hierarchical structure of the folders and data items helps to organize the devices and tags, as shown below.

## 3.2 Devices

In the Configurator, a **device** represents a hardware device that communicates with the OPC server over TCP/IP. A device is directly communicating with its socket, so it is logically under the first level in the **Address Space** tree. Again, the device is represented by its symbolic name. Also, its **IP address** in combination with the **unit identifier** value uniquely identifies the device. It is impossible to have two devices with the same IP address. Setting up a device requires configuring its IP address, unit identifier, TCP port, type, timeouts and optimization parameters.

### 3.2.1 Device Properties

Setting up a device requires configuration of its unique address, type, timeouts and optimization parameters, as shown in the figure below. In the **Name** field, specify a name for the new device.

The screenshot shows the 'Device Properties' dialog box for a device named 'DVTEulator'. The left pane shows the 'Address Space' tree with folders like 'ReservedRegistersBits', 'Simulated', 'InspectionParameters', 'InspectionStatistics', 'Conversions', 'Device Parameters', 'Simulation Signals', and 'Alarm Definitions'. The main pane contains the following fields and sections:

- Name:** DVTEulator
- IP Address:** 127 . 0 . 0 . 1
- Unit Identifier:** 1
- TCP Port:** 502
- ☐ Simulate
- Device Parameters:**
  - ☐ Custom ☒ Default
  - Parameters: <Not Assigned>
- ☒ **DVT Event Alarming**
  - Event Alarming listens for events sent by the DVT sensor on the following port:
  - Event Logging Port: 50000
- Timeouts (ms):**
  - Read: 1000
  - Write: 1000
  - Timeouts to suspend: 3
  - Suspend period: 60000
  - Delay: 20
- Optimizations:**
  - The numbers below indicate how much unused data can be transferred in one message to merge together addresses that are close but not adjacent.
  - Bits: 160
  - Bytes: 10

Buttons at the bottom: Apply, Reset, Add New.

Device Properties

## IP Address

The device **IP Address** is 32-bit value represented as four numbers separated by periods that indicate the location of the device on a TCP/IP network. Each number in the address can range from 0 to 255.

## Unit Identifier

The **Unit Identifier** field may be used to communicate via devices such as bridges and gateways that use a single IP address to support multiple independent end units.

## TCP Port

The default **TCP Port** is 502, which is always a Modbus port.

## Device Parameters

If you have configured device parameters, select the **Custom** radio button and select one of the predefined device types from the **Parameters** drop-down list. For instruction on how to create new or edit device parameters, see the **Device Parameters** section.

## DVT Event Alarming

If the **DVT Event Alarming** check box is checked, the OPC server will detect alarms and events sent by the DVT SmartImage Sensor on the specified **Event Logging Port**. Port 50000 is the default event-logging port for events coming from the sensor.

## Timeouts

Timeout parameters (separately for reading and writing data) specify the period of time (in milliseconds) the server will wait for a response from the device:

- **Read/Write:** Amount of time (in milliseconds) the OPC server will wait for a response (read/write) from the device.
- **Timeouts to Suspend:** The number of consecutive read/write attempts that timeout before the OPC server will suspend communication with the device.
- **Suspend Period:** Amount of time the OPC server will wait before attempting to reconnect to the device.

**Note:** The **Suspend Period** setting can be decreased to reconnect faster with a device that has been restarted, but this will cause the OPC server to generate more network traffic while a device is down.

- **Delay:** Amount of time (in milliseconds) between read attempts.

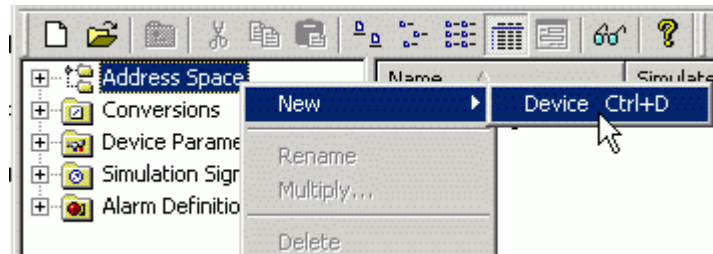
## Optimizations

The server tries to optimize the communication with the devices by requesting as much data as possible in one message. Consecutive registers are merged together into one request for efficiency. The server also can read registers that are not really requested, if this allows it to join two blocks of requested registers. The numbers entered under **Optimizations** specify the maximum block length of adjacent unused data.

### 3.2.2 Creating a New Device

To create a new device:

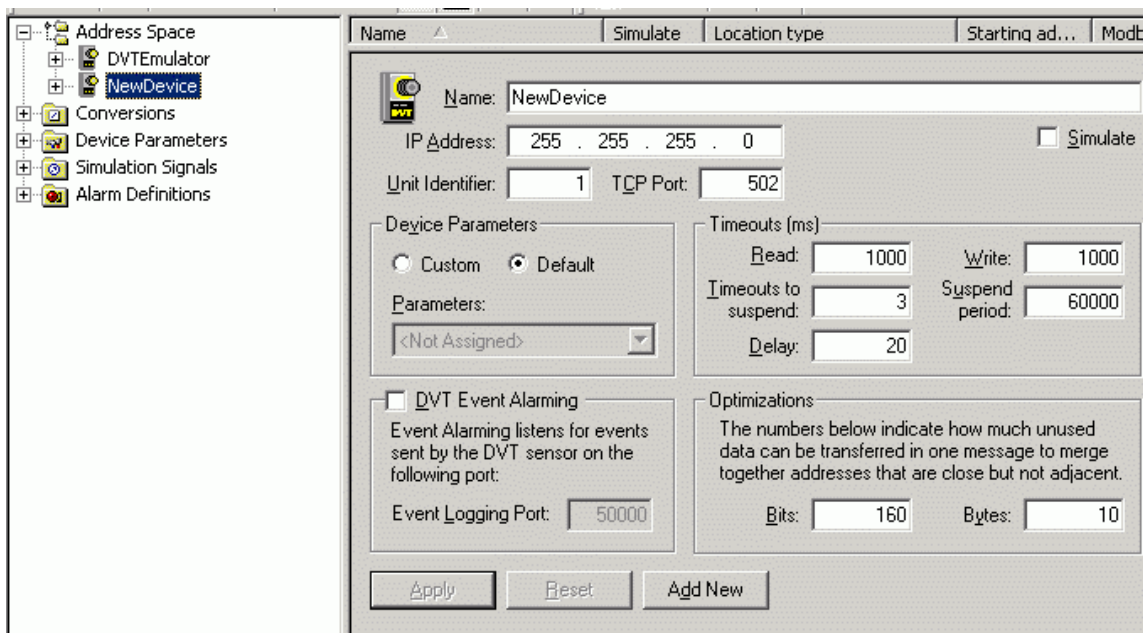
1. Right-click the **Address Space** tree control of the Configurator screen and select **New > Device** from the pop-up menu, as shown in the figure below.



#### Creating a New Device

2. This opens the properties dialog box for the device, as shown in the figure below. In the name field, specify a name for the new device.
3. Configure the properties for the new device. Click the **Apply** button to save your changes.
4. The new device appears under the **Address Space** tree control. Folders containing the inspection statistics and the reserved registers for this device will automatically be created and listed under it.

**Note:** Tags (data items) in the **InspectionParameters** folder, the **InspectionStatistics** folder, and the **ReservedRegistersBits** folder cannot be cut, pasted, copied, or deleted.



#### Setting the Properties for the New Device

### 3.3 Folders

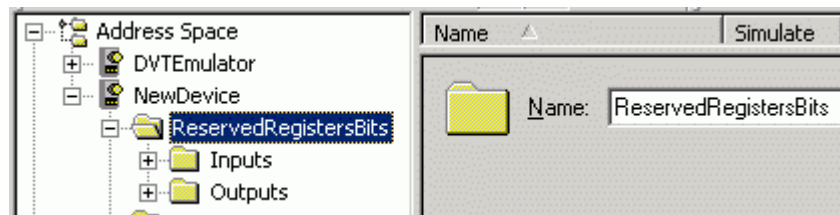
Folders can be used to group items logically. You can configure as many folders as required. Each folder can even have subfolders. The OPC Server supports up to three folder levels. The use of folders is not required.

If desirable, the configuration could just contain data items without any folders. But most likely this will only be useful if the application does not demand too many persistent tags.

#### 3.3.1 Folder Properties

For each new device created, the Configurator automatically creates a folder called **ReservedRegistersBits**, which contains a predefined set of data items (tags), as well as folders called **InspectionParameters** and **InspectionStatistics**, which contain inspection tags. In the folder properties, you can change the name of the folder, as shown in the figure below.

**Note:** Tags (data items) in the **InspectionParameters** folder, the **InspectionStatistics** folder, and the **ReservedRegistersBits** folder cannot be cut, pasted, copied, or deleted.

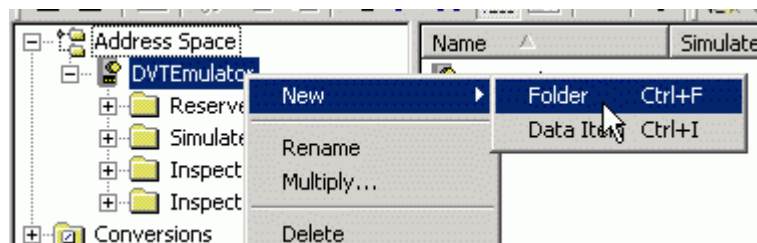


Folder Properties

#### 3.3.2 Creating a New Folder

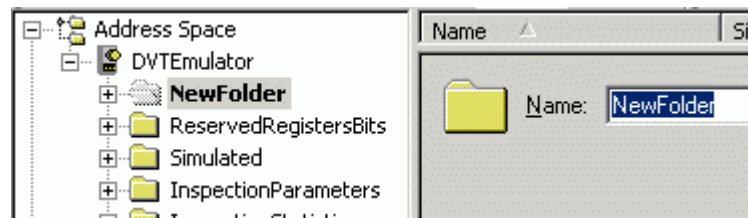
To create a new folder:

1. Right-click a device or folder on the tree control of the Configurator screen and select **New > Folder** from the pop-up menu, as shown in the figure below.



Creating a New Folder

2. This opens the properties dialog for the folder, as shown in the figure below. In the name field, specify a name for the new folder.
3. Click the **Apply** button to save your changes.
4. The new folder appears under the tree control.



Naming the New Folder

### 3.4 Data Items

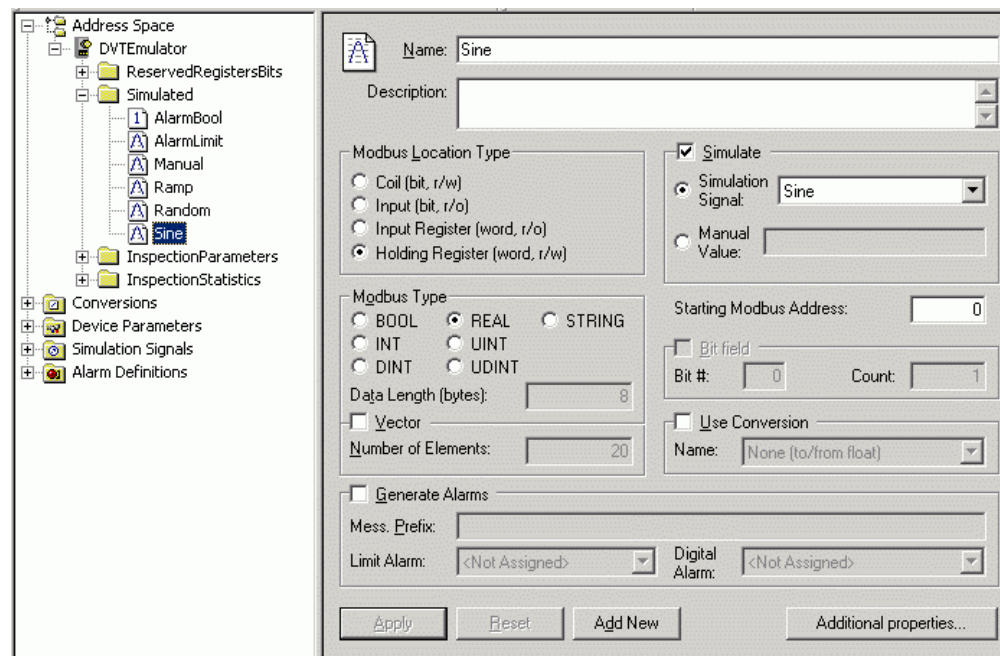
A **Data Item** represents a register in the device or a range of registers. A symbolic name and description is associated with the data item. An OPC client can obtain the data item description.

The actual OPC item name (tag) is compounded from the **Address Space** root, the names of the folder and its subfolders, and the name of the data item. Data items can be located in any folder, even in the root of the address space.

**Note:** Tags (data items) in the **InspectionParameters** folder, the **InspectionStatistics** folder, and the **ReservedRegistersBits** folder cannot be cut, pasted, copied, or deleted. Most tag properties for reserved registers cannot be modified. However, you can change alarm information, configure **Additional Properties**, and modify the **Description** field for reserved registers.

#### 3.4.1 Data Item Properties

Configure the following settings for a data item, as shown in the figure below.



Data Item Properties

- **Name:** A logical name for the data item (Setpoint, Param001, ON\_OFF, etc).
- **Description:** A descriptive comment for the data item.
- **Location Type:** Location type is a type of a register in the device. Every device is identified by its unique address. Its registers are read as Input (1 bit long) or Input Register (16 bits), or written to as Coil (1 bit) or Holding Register (16 bits). Registers of each type are addressed by using 16-bit numbers. Device registers are divided into **Coils, Inputs, Input Registers** and **Holding Registers**. The table below explains the name conventions used:

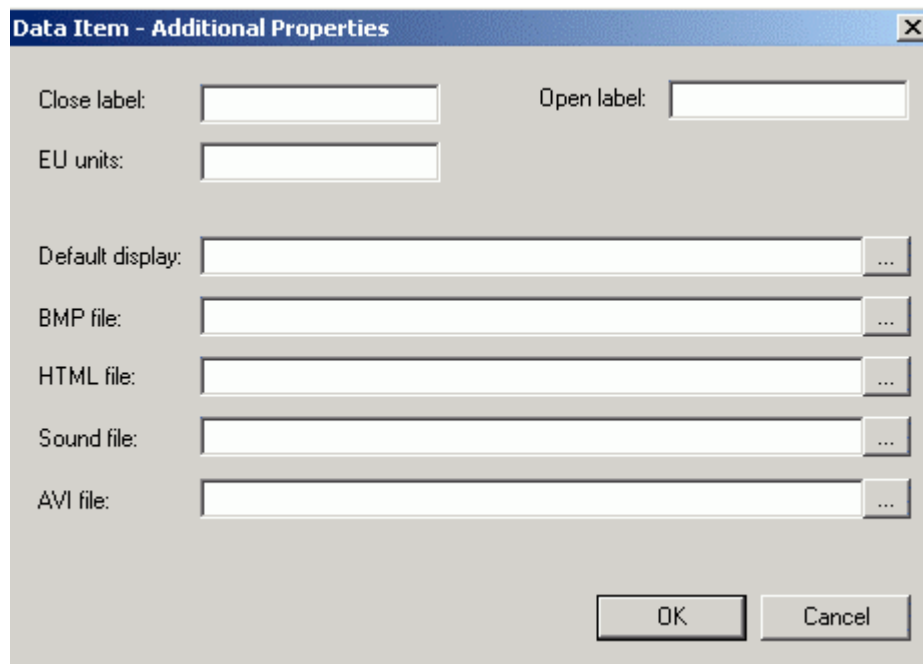
|        | <b>Read Only</b> | <b>Read / Write</b> |
|--------|------------------|---------------------|
| 1 bit  | Input            | Coil                |
| 16 bit | Input Register   | Holding Register    |

- **Modbus Type:** The location type (device data) will be understood as Modbus type (OPC data type). Modbus data type also depends on the Location type selected. The **Data Length (bytes)** field tells you how long (in bytes) the field is for the Modbus type (e.g. REAL, INT, UINT, etc.) selected. The BOOL length is always 1 bit.  
**Example:** Coil or Input (1 bit) device data type can be Modbus BOOL only. When selecting Modbus STRING type, you must specify the data length (how many bytes will the STRING be represented by).
- **Simulation:** To test the client functionality, choose a **Simulation Signal** from the **Simulation Signal** drop-down list and check the **Simulate** check box. See the **Simulation Signals** section for information about creating simulation signals. All levels in the Address Space (port, device, folder, data item) support the process of simulation (**Simulate** check box). The parent list in the tree is superior; it has a higher priority when deciding to simulate the data item or not. In other words, a data item is simulated, if it itself has a simulation selected, or if any of its parents has the **Simulate** check box checked. (It may be simulated even if its **Simulate** check box stays unchecked.)
- **Manual Value:** If the **Simulate** check box is checked, the data item will offer a constant parameter value, because the **Manual Value** setting is of the highest priority. The changes in the configuration take effect only when the server reloads the configuration (on startup).
- **Starting Modbus Address:** This value specifies the data item address (register number) in the device data space. With the UINT Modbus type, it is possible to extract bits from the register and use them as a Boolean or integer value (this functionality is read-only). You can specify a group of **Count** adjacent bits inside a word starting with **Bit #**. This way, it is possible to use a register for several separate data items.
- **Use Conversion:** To get the data value converted according to a prescribed form, choose one of the predefined or user-defined conversions. See **Chapter 6** for more details.
- **Generate Alarms:** Check the **Generate Alarms** check box to make the server generate alarms based on the data item value. The **Message Prefix** parameter is the text of the message for this data item; it will be followed by the text configured for a particular alarm type. The second part of the alarm message will contain the **Message Body** string (see **Alarm Definitions**). The

server allows having any number of alarm definitions (templates) predefined. You can then combine one of them with the specific tags.

### Additional Properties

Clicking the **Additional properties** button opens the **Additional Properties** dialog box, shown below, which allows you to set a textual string for an Open/Close label, a Default display, a .bmp file, an HTML file, a sound file, and an .avi file. These properties are also accessible for OPC Data Access clients. For example, to see the HTML file name being presented in the OPC client, you should append **.HTMLFile** to the OPC Item name.

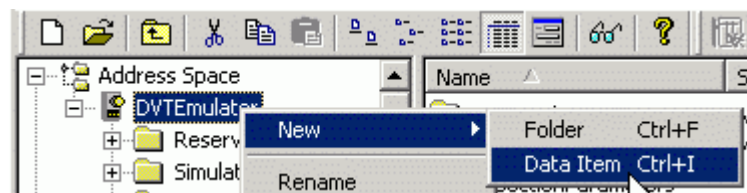


**Additional Properties Dialog Box**

### 3.4.2 Creating a New Data Item

To create a new data item:

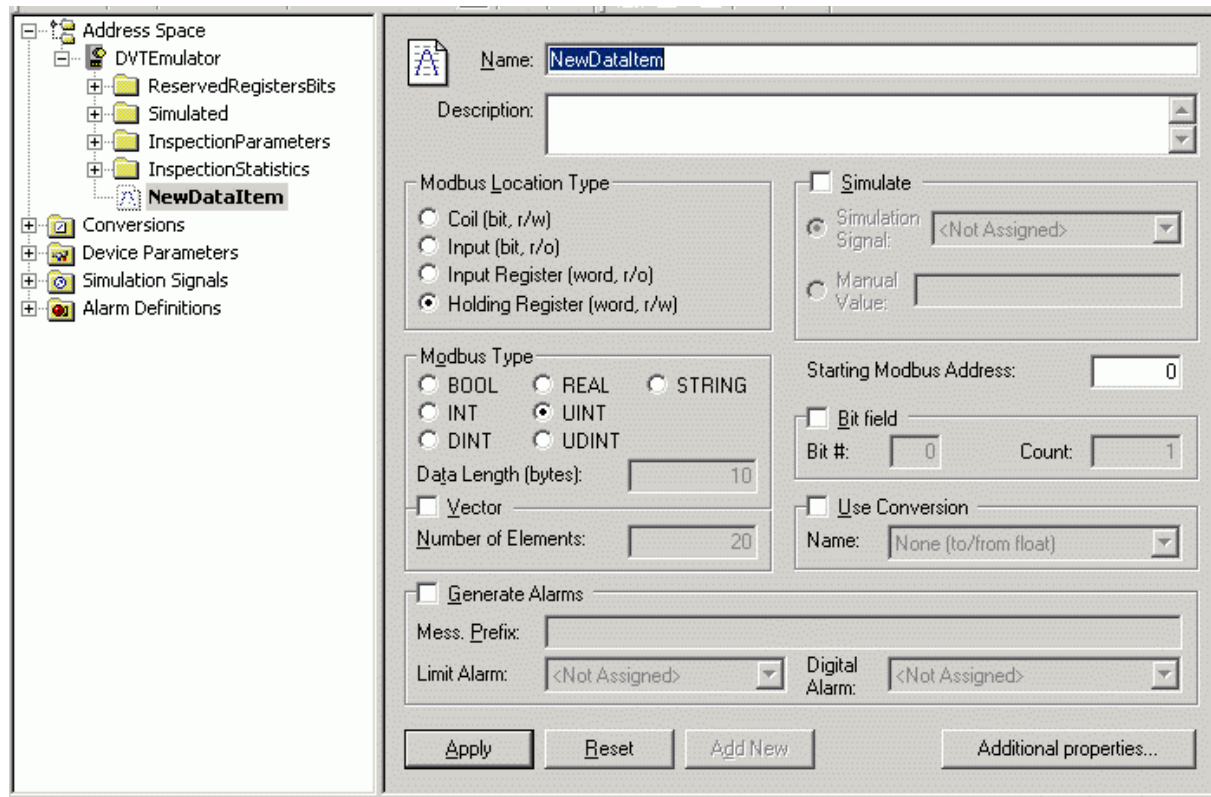
1. Right-click a device or folder on the tree control of the Configurator screen and select **New > Data Item** from the pop-up menu, as shown in the figure below.



**Creating a New Data Item**

2. This opens the properties dialog box for the data item, as shown in the figure below. In the name field, specify a name and a description for the new data item.

3. Configure the properties for the new data item. Click the **Apply** button to save your changes.
4. The new data item appears under the tree control.



Setting the Properties for the New Data Item

# Alarm Definitions

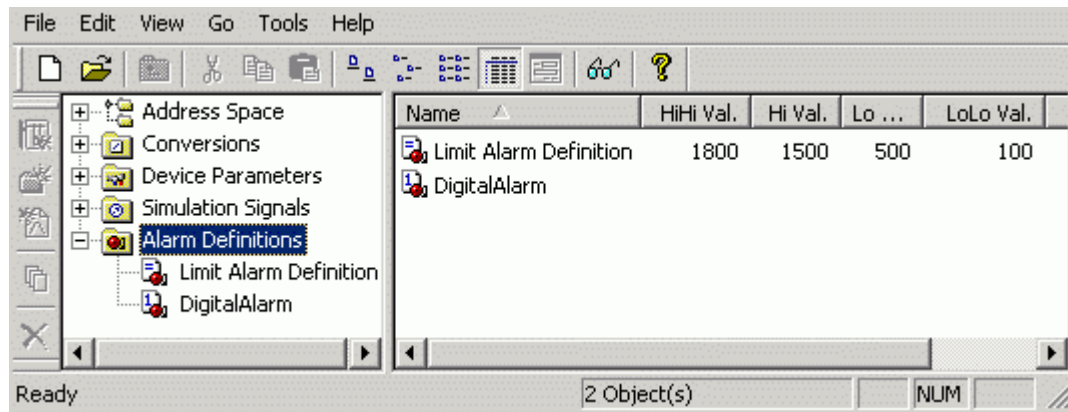
# 4

## 4.1 Configuring Alarm Definitions

The DVT OPC Server is an OPC Alarm and Events server that supports the following alarm types:

- **Limit** (High High, High, Low, Low Low)
- **Digital**

Alarm parameters are set up in the Configurator under the **Alarm Definitions** tree control, shown in the figure below, which is divided into two alarm template types: **Digital** alarm definitions and **Limit** (analog) alarm definitions. Digital alarms can be defined for a data item of "BOOL" type only, while Limit alarms can be defined for all other types except for the "String" data type. Default alarm definitions are provided in the sample configuration database.



Alarm Definitions Tree Control

## 4.2 Limit Alarm Definitions

A limit alarm sets the values for four levels of alarms: **LoLo**, **Lo**, **Hi**, and **HiHi**. Limit alarm parameters can have subranges within the data item amplitude. Every subrange definition includes a **Message Body** that will be appended to the alarm message, the **Severity** of the alarm, and the **Req. Ack.** flag for alarm acknowledgement.

### 4.2.1 Limit Alarm Definition Properties

In the **Limit Alarm Definition Properties** dialog box, shown below, configure the following settings:

#### Limit Alarm Definition Properties

- In the **Name** field, type a name for the new limit alarm.
- **Update Rate** defines the frequency of checking the data item value (in milliseconds), and possibly responding by sending the alarm message.
- **Deadband** prevents the server from generating a huge amount of alarm messages and overloading the clients when the signal oscillates around one of the limits specified. The deadband value extends the limit zone. It results in sending only one alarm message even if the signal oscillates. Deadband indicates the deadband value to apply to the converted analog values. The deadband value is required and is calculated on borderline alarming limit values to prevent repeated alarm cycles.
- The **Value** field is used to calculate the state of the input fields. For example, a value of 10 for LoLo is compared with the value of the input to determine if the alarm is in LoLo state.
- In the **Message Body** field, enter the warning message that will appear when the alarm is sent. The message can be any text string.
- In the **Return to Normal** field, check the check box and then enter the text that will appear when the alarm is taken care of (e.g. has been acknowledged). The message can be any text string.

**Note:** It is not necessary to enter a message text or a base text. The Server will default to the OPC subcondition name and the OPC condition name. For example, a LoLo alarm will post a description of LoLo limit.

- **Severity** is the OPC-defined value for alarm Priority. The valid OPC severity range is 0 (lowest) to 1000 (highest).

- The **Requires Ack** field is used for OPC condition alarms to determine whether the alarm needs a user acknowledgement. If the **Requires Ack** field is checked, then the alarm requires a user acknowledgement. If the value is not checked, then the alarm is posted as already acknowledged.

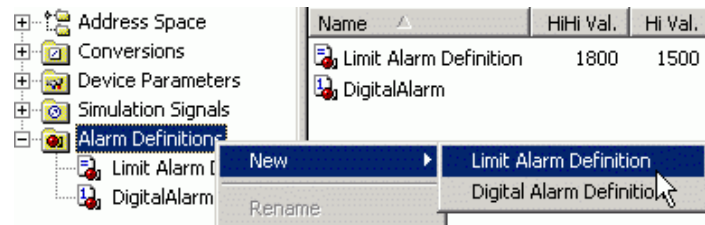
**Note:** Changes to the alarm property fields (HiHi, LoLo, Hi, Lo, Message Text, etc.) in runtime through an OPC tag update will be automatically saved to the database, over-writing any values specified in configuration mode.

- **Apply:** Saves all changes specified in the properties dialog box. The alarm definition appears in the tree control.
- **Reset:** Restores the default limit alarm definition settings.
- **Add New:** Clicking this button starts configuration of a new alarm definition.

#### 4.2.2 Creating a New Limit Alarm Definition

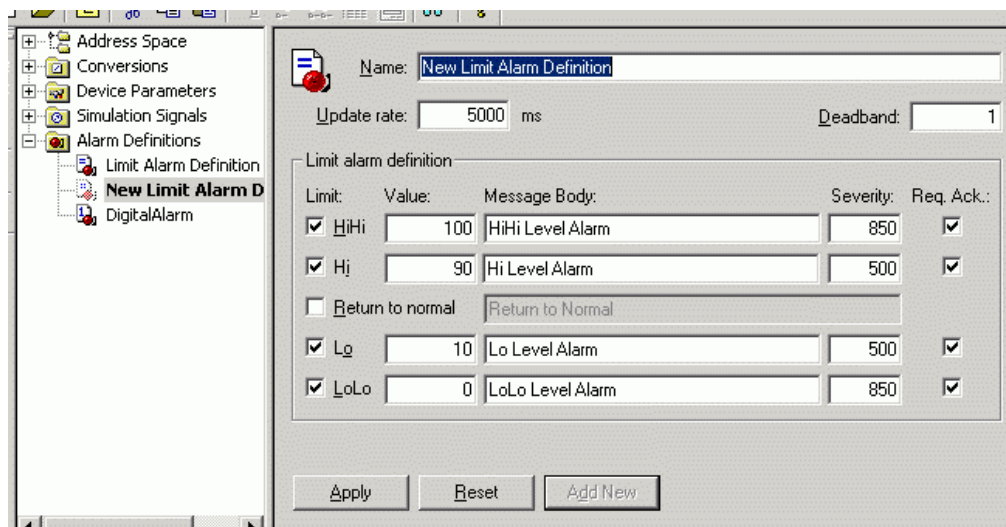
To create a new limit alarm definition:

1. Right-click the **Alarm Definitions** folder on the tree control of the Configurator screen and select **New > Limit Alarm Definition** from the pop-up menu, as shown in the figure below.



Creating a New Limit Alarm Definition

2. The **Limit Alarm Definition Properties** dialog appears, as shown in the figure below.



Setting the Properties for the New Limit Alarm Definition

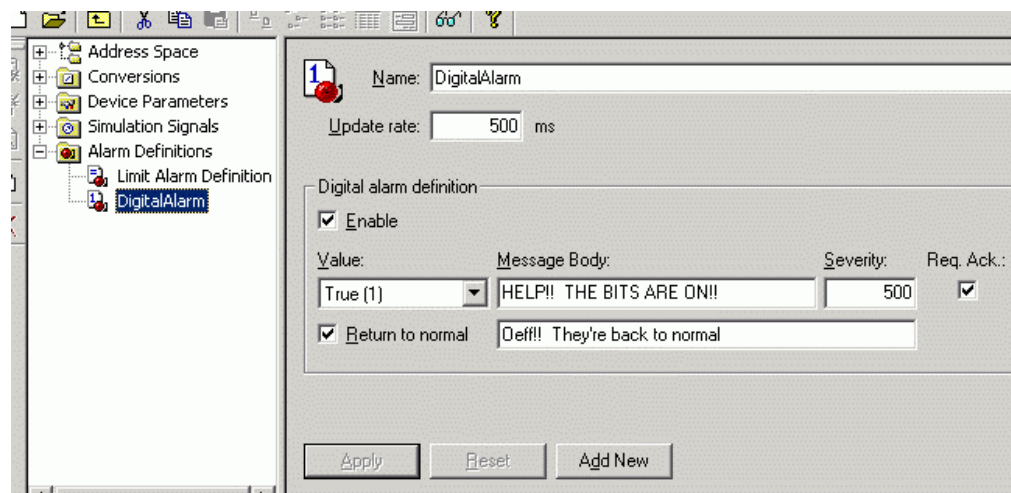
3. In the **Name** field, type a name for the new alarm definition.
4. Specify the limits, values, and severity levels for the alarm.
5. Specify a server **Update Rate** (in milliseconds) as well as a **Deadband** value.
6. When you have finished configuring the alarm definition properties, click the **Apply** button. The new configuration appears under the **Alarm Definitions** tree control.

## 4.3 Digital Alarm Definitions

A digital alarm sets an alarm if the comparison between the **Alarm State Value** and the input state is TRUE.

### 4.3.1 Digital Alarm Definition Properties

In the **Digital Alarm Definition Properties** dialog box, shown below, configure the following settings:



**Digital Alarm Definition Properties**

- In the **Name** field, type a name for the new digital alarm.
- **Update Rate** defines the frequency of checking the data item value (in milliseconds), and possibly responding by sending the alarm message.
- Specify a **Value** for the digital alarm (True or False).
- In the **Message Body** field, enter the warning message that will appear when the alarm is sent. The message can be any text string.
- In the **Return to Normal** field, check the check box and then enter the text that will appear when the alarm is taken care of (e.g. has been acknowledged). The message can be any text string.

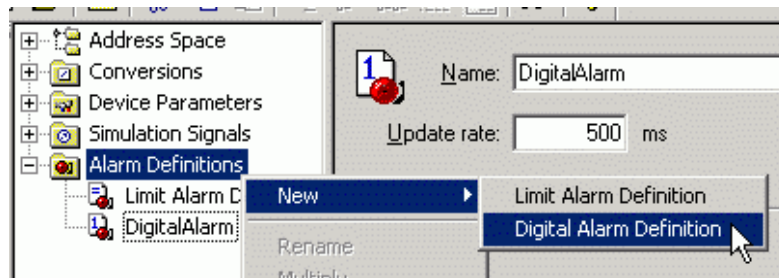
**Note:** It is not necessary to enter a message text or a base text. The Server will default to the OPC subcondition name and the OPC condition name.

- **Severity** is the OPC-defined value for alarm Priority. The valid OPC severity range is 0 (lowest) to 1000 (highest).
- The **Requires Ack** field is used for OPC condition alarms to determine whether the alarm needs a user acknowledgement.
- **Apply:** Saves all changes specified in the properties dialog box. The alarm definition appears in the tree control.
- **Reset:** Restores the default digital alarm definition settings.
- **Add New:** Clicking this button starts configuration of a new digital alarm definition.

### 4.3.2 Creating a New Digital Alarm Definition

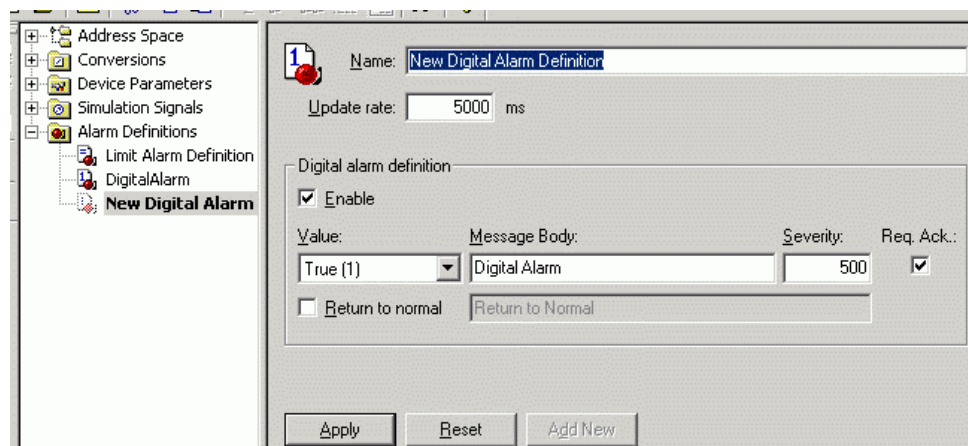
To create a new digital alarm definition:

1. Right-click the **Alarm Definitions** folder on the tree control of the Configurator screen and select **New > Digital Alarm Definition** from the pop-up menu, as shown in the figure below.



Creating a New Digital Alarm Definition

2. The **Digital Alarm Definition Properties** dialog appears, as shown in the figure below.



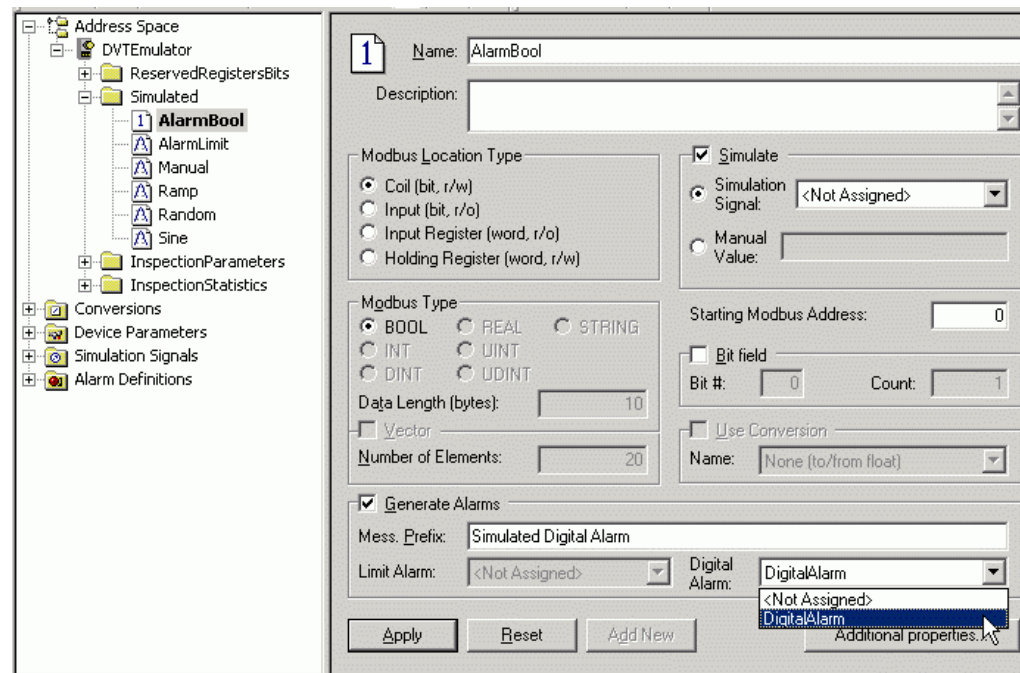
Setting the Properties for the New Digital Alarm Definition

3. In the **Name** field, type a name for the new alarm definition.
4. Specify the value and severity level for the alarm.
5. Specify a server **Update Rate** (in milliseconds).
6. When you have finished configuring the alarm definition properties, click the **Apply** button. The new configuration appears under the **Alarm Definitions** tree control.

## 4.4 Assigning Alarm Definitions to Tags

Alarm definitions can be associated with data tags via the **Tag Properties** dialog, as shown in the figure below. In the tag properties, select an alarm definition from the drop-down list, as shown in the figure below.

**Note:** The type of alarm that can be assigned to a particular tag is restricted by the data type of the tag. For instance, Boolean tags can only be assigned a digital alarm. Inspection statistics tags can only be assigned limit alarms because they are not Boolean values.



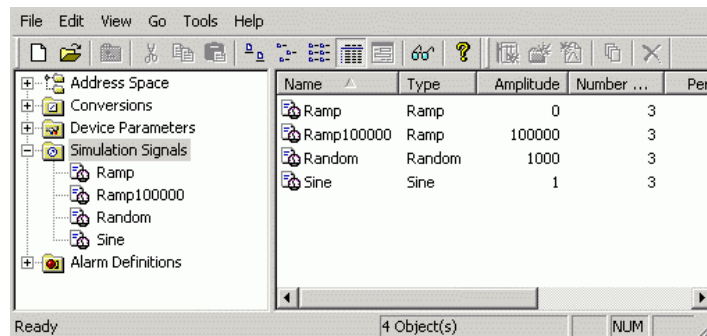
Assigning Alarm Definitions to Tags

# Simulation Signals

5

## 5.1 Configuring Simulation Signals

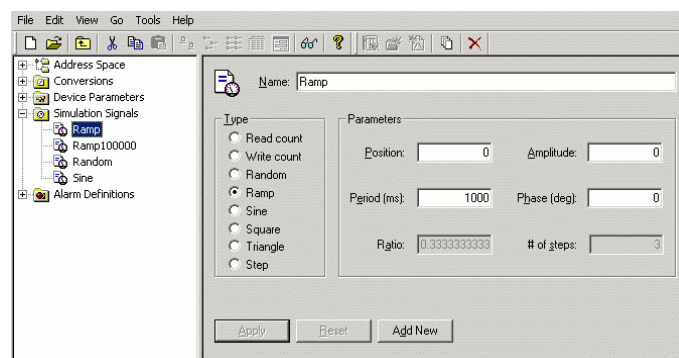
The Configurator offers various OPC data items in the **Simulation Signals** tree control, as shown in the figure below. To test the client functionality, choose a simulation signal from the tree control. All levels in the **Address Space** folder support the process of simulation. The parent list in the tree control is superior; it has higher priority when deciding to simulate the data item or not.



Simulation Signals Tree Control

### 5.1.1 Simulation Signal Properties

In the simulation signal properties, shown in the figure below, configure the following settings:



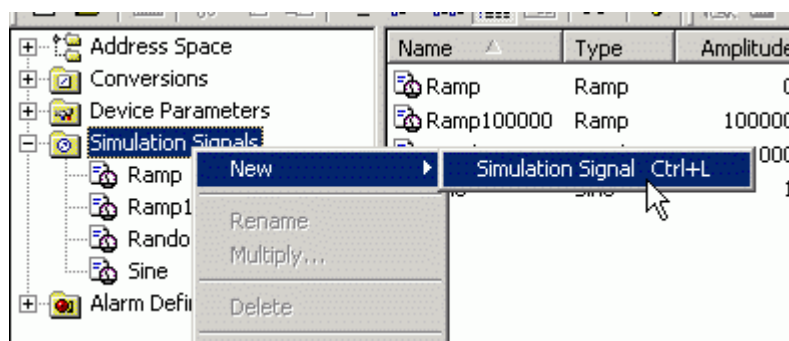
Simulation Signal Properties

- **Name:** Specifies the name of the selected simulation signal. The name can be up to 12 alphanumeric characters, including underscores ( \_ ) and hyphens ( - ).
- **Signal Type:** For each signal, you can select one of the following signal types from the drop-down list:
  - \* **Read Count** is incremented by one every time when the item is read.
  - \* **Write Count** increments when the item is written.
  - \* **Random** generates random value within the **Amplitude** range starting with **Position**.
  - \* **Ramp, Sine, Square, Triangle** and **Step** are periodical signals. Their time behavior is influenced by **Period** and **Phase** parameters. **Period** specifies the signal frequency (in milliseconds), while **Phase** moves the signal origin on the time axis (in degrees).
  - \* **Square** and **Triangle** have one additional parameter: **Ratio**. Ratio defines Triangle signal steepness, or Square signal H/L proportions.
- The **Number of Steps** parameter of the **Step** signal defines the number of steps into which the signal amplitude will be divided.
- **Apply:** Saves all changes specified in the properties dialog box. The simulation signal appears in the tree control.
- **Reset:** Restores the default simulation signal settings.
- **Add New:** Clicking this button starts configuration of a new simulation signal.

### 5.1.2 Creating a New Simulation Signal

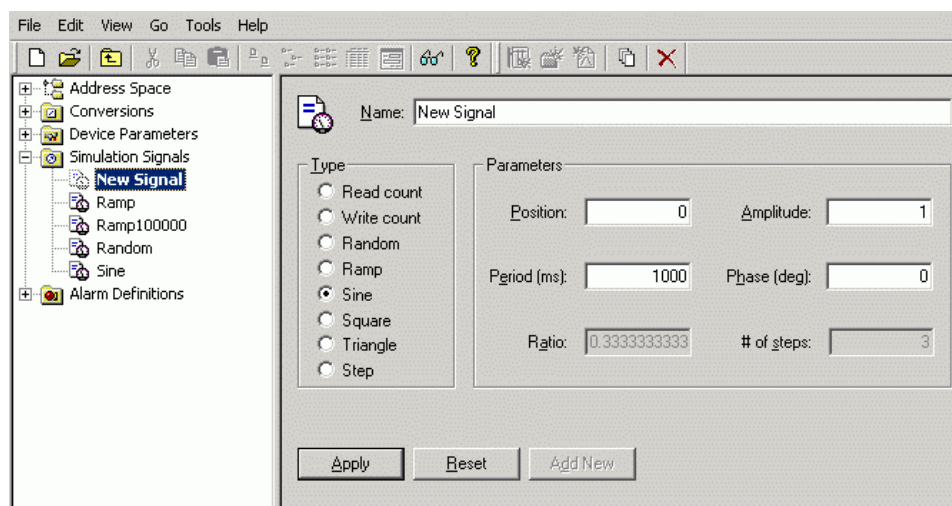
To create a new simulation signal:

1. Right-click the **Simulation Definitions** tree control and select **New > Simulation Signal** from the pop-up menu, as shown in the figure below.



Creating a New Simulation Signal

2. The properties dialog box for the new configuration appears in the right-hand pane of the Configurator, as shown in the figure below.



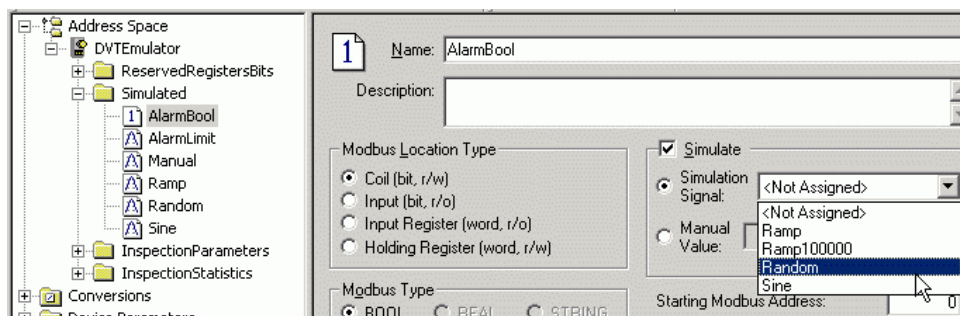
**Setting the Properties for the New Simulation Signal**

3. In the **Name** field, type a name for the new signal.
4. Select a **Signal Type**.
5. When you have finished configuring the simulation signal properties, click the **Apply** button. The new configuration appears under the **Simulation Definitions** tree control.

### 5.1.3 Assigning Simulation Signals to Tags

Simulation signals can be associated with data tags via the **Tag Properties** dialog, as shown in the figure below. In the tag properties, select a simulation signal from the drop-down list, as shown in the figure below.

**Note:** Simulation signals cannot be assigned to reserved registers or inspection parameters.

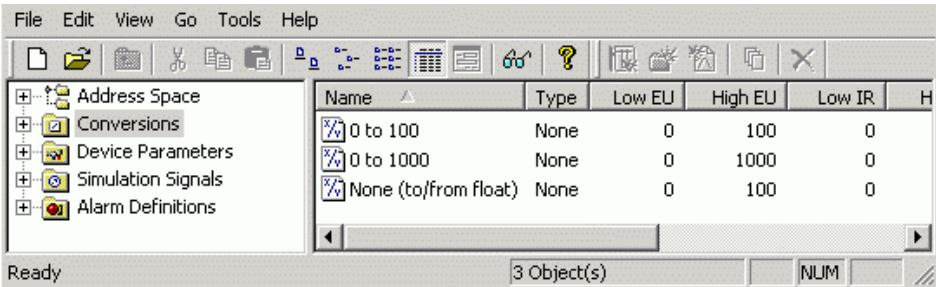


**Assigning Simulation Signals to Tags**

# Conversions and Device Parameters 6

## 6.1 Configuring Conversion Definitions

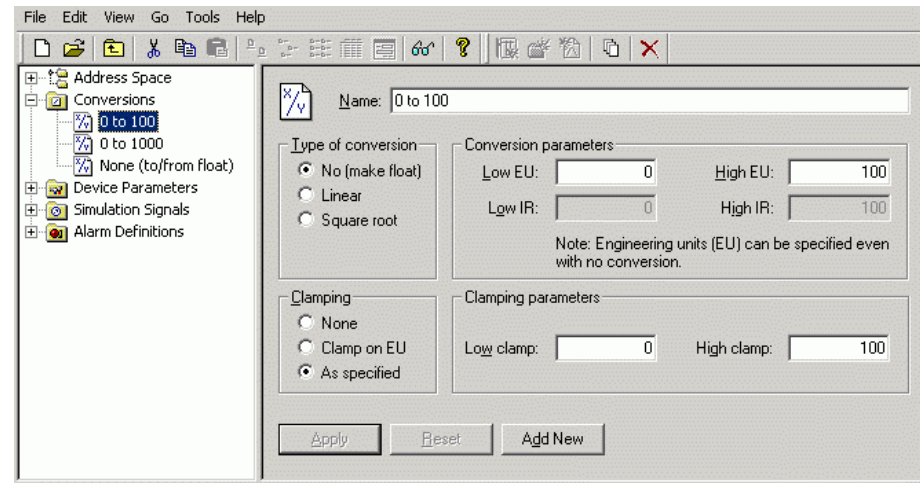
You can tell the server to convert device data values simply by setting the **Conversion** properties in the **Conversions** tree control, as shown in the figure below. Conversion settings can be associated with a data tag via the tag properties.



Conversions Tree Control

### 6.1.1 Conversion Properties

In the conversion properties, shown in the figure below, configure the following conversion settings for data tags:



Conversions Properties

- **Name:** Specifies the name of the conversion definition. The name can be up to 12 alphanumeric characters, including underscores ( \_ ) and hyphens ( - ).
- **Conversion Type:** For each conversion definition, select one of the following conversion types from the drop-down list:
  - \* **None (make float):** Converts the data into float data type, but does not change the value itself. When this option is selected, the **Engineering Units** and **Instrument Range** fields are disabled.
  - \* **Linear:** Keeps a linear relation between EU and IR.
  - \* **Square Root:** Keeps a square root relation between EU and IR.
- **Engineering Units (EU):** Client scale; specify low and high values for the engineering units (if applicable).
- **Instrument Range (IR):** Device scale; specify low and high values for the instrument range (if applicable).
- **Clamping:** If clamping is enabled, the data value will be limited to its High Value/EU High Value when it exceeds the upper limit, and similarly to its Low Value/Low EU Value parameter when it exceeds the bottom limit. Select one of the following clamping types from the drop-down list:
  - \* **None:** No clamping type is specified.
  - \* **Clamp on EU:** Clamps on the specified low and high engineering units (EU) values.
  - \* **Specified:** Clamps on a specified range of low and high values.
- **Range:** Specify low and high values for the range.
- **Apply:** Saves all changes specified in the properties dialog box. The conversion definition appears in the tree control.
- **Reset:** Restores the default conversion definition settings.
- **Add New:** Clicking this button starts configuration of a new conversion definition.

### 6.1.2 Creating a New Conversion

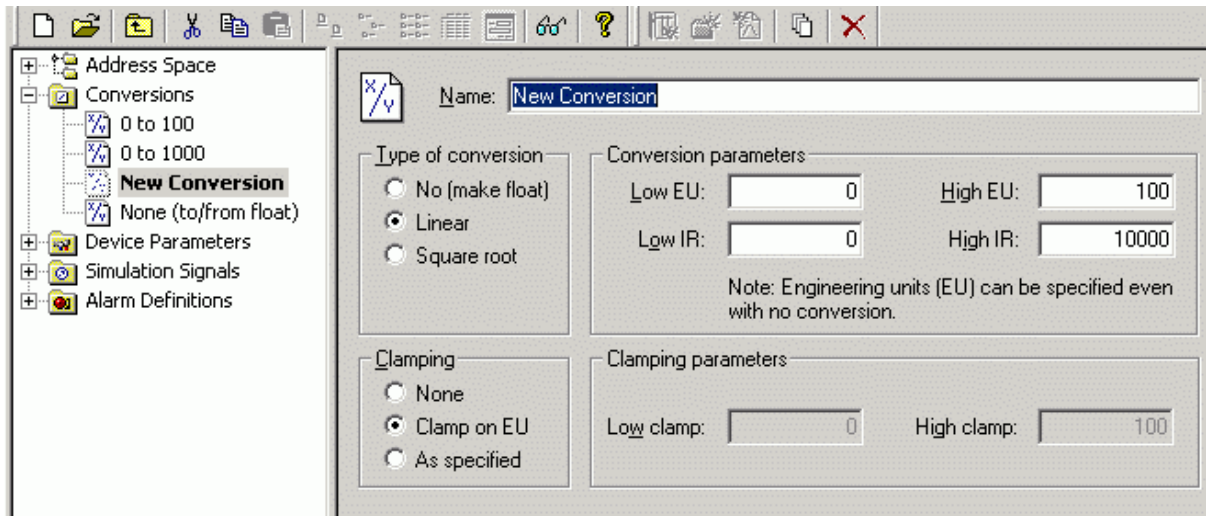
To create a new conversion:

1. Right-click the **Conversions** tree control and select **New > Conversion** from the pop-up menu, as shown in the figure below.



Creating a New Conversion

2. The properties dialog box for the new configuration appears in the right-hand pane of the Configurator, as shown in the figure below.



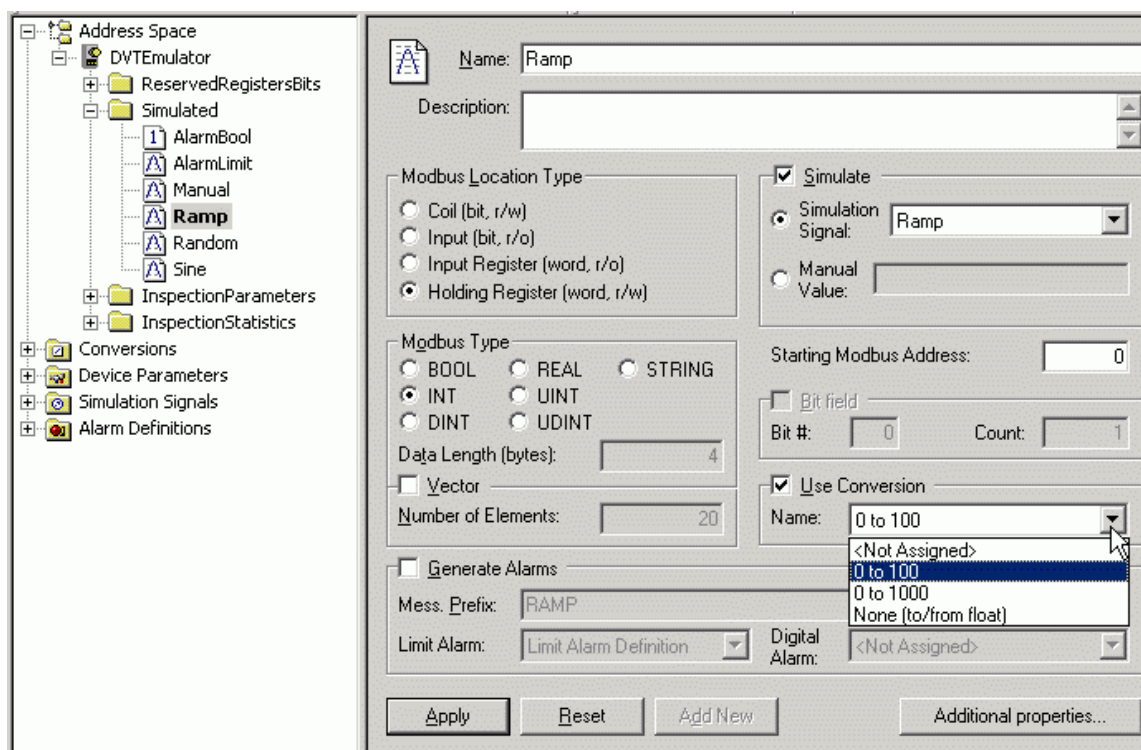
### Setting the Properties for the New Conversion

3. In the **Name** field, type a name for the new conversion.
4. Choose a **Conversion Type** and specify low and high values for engineering units and instrument range (if applicable).
5. Choose a **Clamping Type** and specify low and high values for the range.
6. When you have finished configuring the conversion properties, click the **Apply** button. The new configuration appears under the **Conversions** tree control.

### 6.1.3 Assigning Conversion Definitions to Tags

Conversion settings can be associated with data tags via the **Tag Properties** dialog, as shown in the figure below. In the tag properties, check the **Use Conversion** check box and select a conversion from the drop-down list, as shown in the figure below.

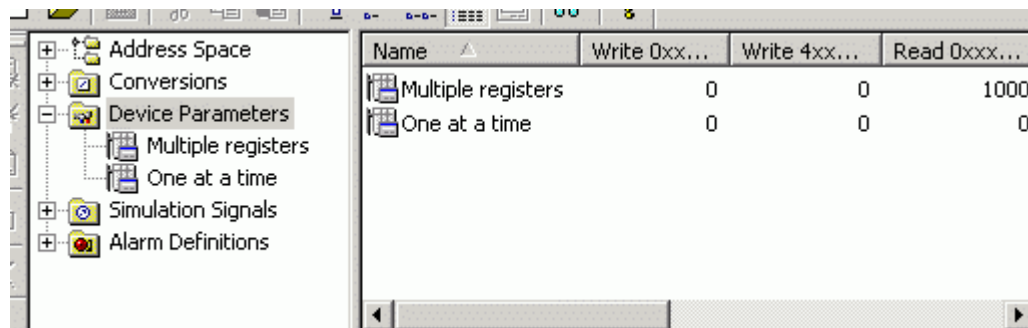
**Note:** Conversion definitions cannot be applied to reserved registers or inspection parameters.



Assigning Conversion Definitions to Tags

## 6.2 Configuring Device Parameters

The **Device Parameters** tree control, shown in the figure below, contains a list of custom device types. Device parameters influence the behavior and performance of the server for the device.

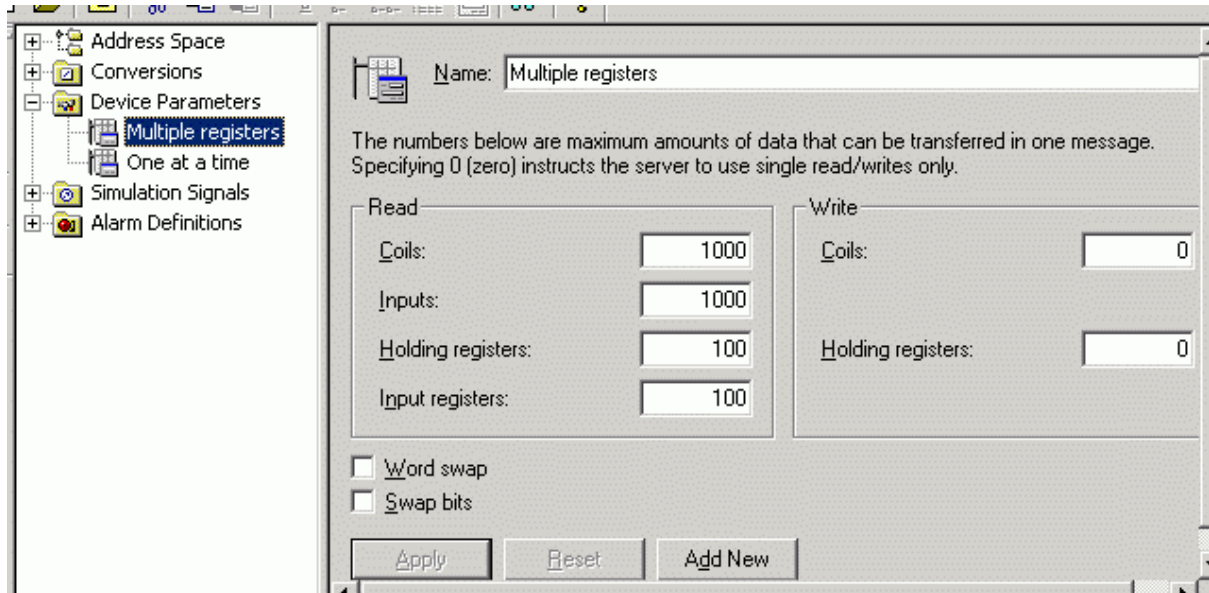


Device Parameters Tree Control

### 6.2.1 Device Parameter Properties

Every device is identified by its unique address. Its registers are read as Input (1 bit long) or Input Register (16 bits), or written to as Coil (1 bit) or Holding Register (16 bits). Registers of each type are addressed by using 16-bit numbers.

The meaning of numbers in the device parameters dialog, shown in the figure below, is the maximum amount of data that can be transferred in one message. Setting the value equal to 0 (zero) forces the server to use single read/write messages only.



**Device Parameter Properties**

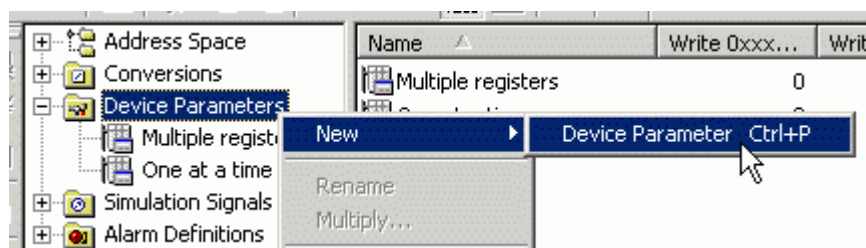
You can also configure the following additional properties for each device type:

- **Word swap:** Swaps the first word with the second when reading/writing DINT, UDINT or float values. Word swapping is checked by default to make it consistent with the DVT way of storing multi-byte variables.
- **Reverse bits:** Reverses the order of bits in word-sized values (least significant bit becomes the most significant bit).
- **Apply:** Saves all changes specified in the properties dialog box. The device parameter appears in the tree control.
- **Reset:** Restores the default device parameter settings.
- **Add New:** Clicking this button starts configuration of a new device parameter.

## 6.2.2 Creating a New Device Parameter

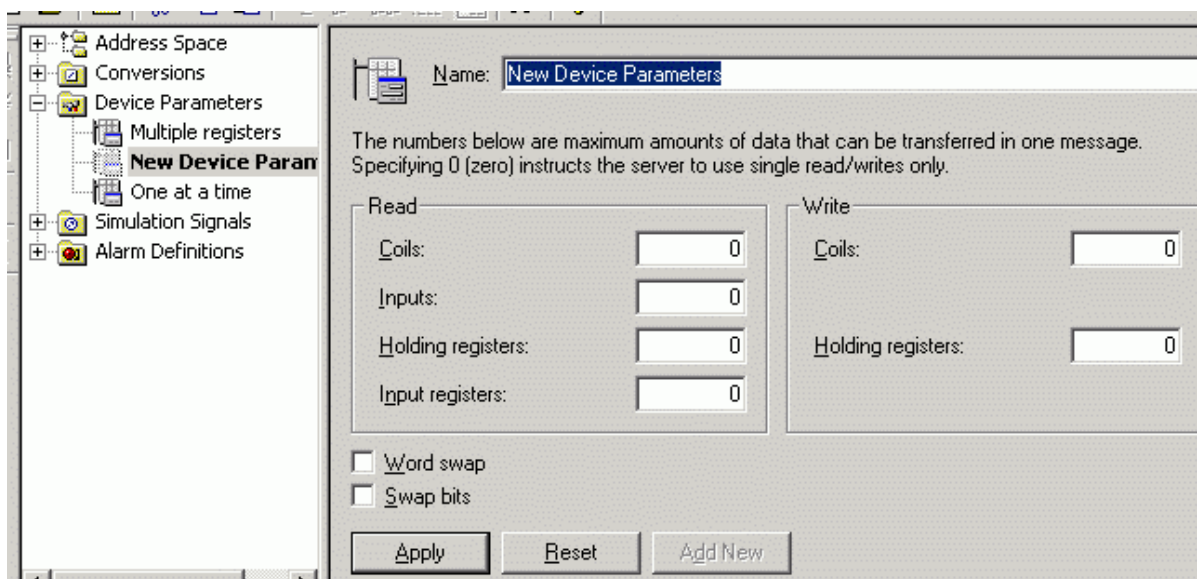
To create a new device parameter:

1. Right-click the **Device Parameters** tree control and select **New > Conversion** from the pop-up menu, as shown in the figure below.



### Creating a New Device Parameter

2. The properties dialog box for the new configuration appears in the right-hand pane of the Configurator, as shown in the figure below.

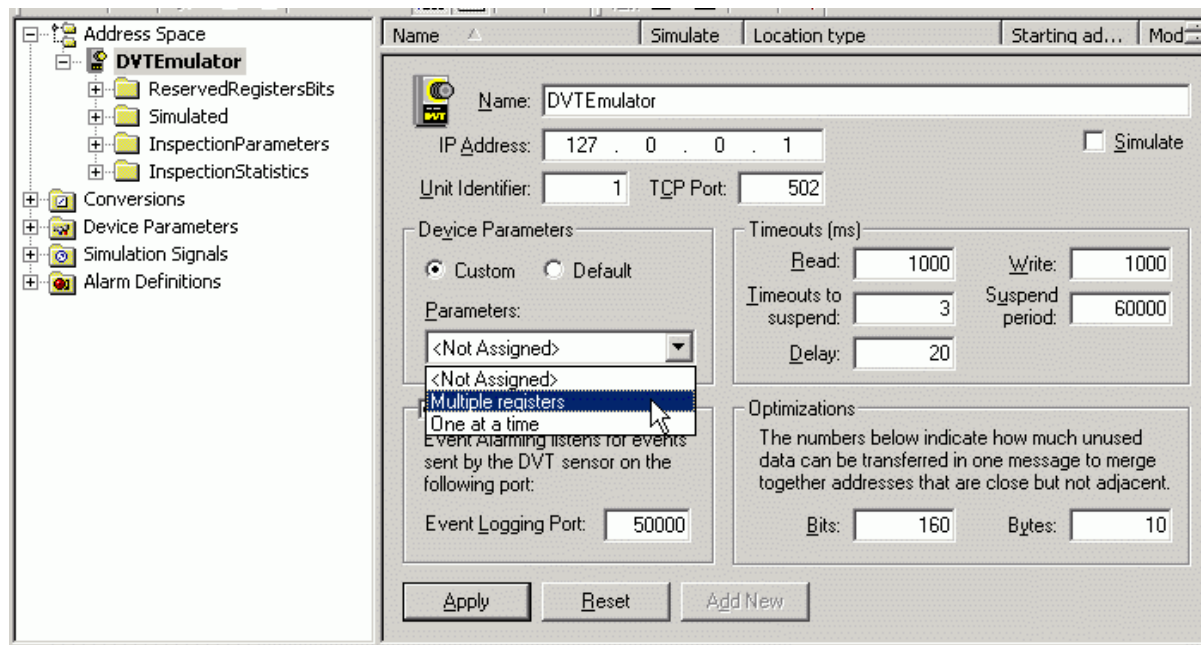


### Setting the Properties for the New Device Parameter

3. In the **Name** field, type a name for the new device parameter.
4. Specify the maximum amounts of data that can be transferred in one message. Setting the value equal to 0 (zero) forces the server to use single read/write messages only.
5. When you have finished configuring the device parameter properties, click the **Apply** button. The new configuration appears under the **Device Parameters** tree control.

### 6.2.3 Assigning Device Parameters to Devices

Device parameters can be associated with devices via the **Device Properties** dialog, as shown in the figure below. In the device properties, check the **Custom** radio button and select a device parameter from the drop-down list, as shown in the figure below.



Assigning Device Parameters to Devices

# Runtime Operations

# 7

## 7.1 DVT OPC Server Runtime Module

The runtime module of the OPC Server gets started by the operating system when an OPC client tries to create an instance of the server. Upon startup, the server loads the currently active configuration created with the OPC Server Configurator. Runtime mode makes accessible items defined in the active configuration database. Also, some new items are created automatically (depending on the type).

The runtime component of the DVT OPC Server will be automatically started by attempting a client connection to the server through a suitable OPC client application.

**Note:** Once the client is connected and the OPC server is in runtime mode, any configuration changes will take about 10 seconds to update in the server.

### Key Runtime Features of DVT OPC Server

The key runtime features of DVT OPC Server include:

- DVT reserved registers read/write.
- DVT inspection statistics read/write.
- Alarm-and-event logging support.
- TraceWorX error-logging support.
- Runtime monitor for viewing runtime statistics.
- Can run as a service or an executable.

## 7.2 DVT Reserved Registers

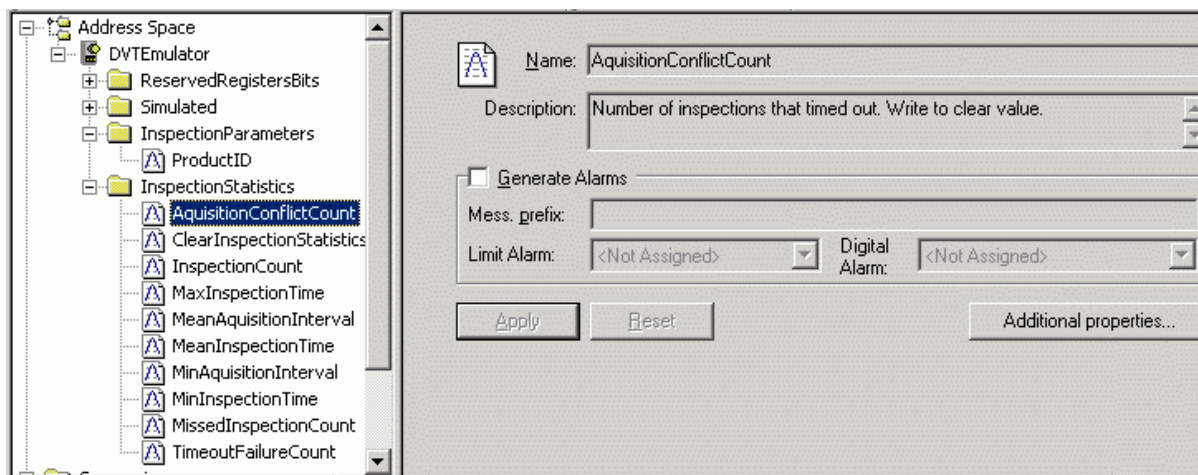
The majority of the data transfer between the OPC server and the DVT SmartImage Sensors will take place over the Modbus protocol. The sensors support the Modbus protocol by default through an array of DVT registers, which can be mapped to Modbus registers. For a list of **reserved registers** that can be used to monitor and control the functioning of the sensors, please refer to **Chapter 1**.

### 7.3 Inspection Statistics

The DVT OPC Server supports read/write operations for DVT inspection statistics. The OPC server provides the ability to trigger the sensors to execute inspections from an OPC client. One way to do this is to use the Trigger bit located in the DVT reserved registers. By setting and then clearing the Trigger bit, a toggle effect is simulated and an inspection is triggered.

Each device created under the **Address Space** tree control in the DVT OPC Server Configurator includes a predefined set of inspection tags, which are located by default in the **InspectionParameters** and **InspectionStatistics** folders. In the tag properties, you can assign alarm definitions to an inspection tag as well as configure **Additional Properties** for the tag, as shown in the figure below. For information about alarm definitions, please refer to **Chapter 4**. For information about the **Additional Properties** dialog box, please refer to **Chapter 3**.

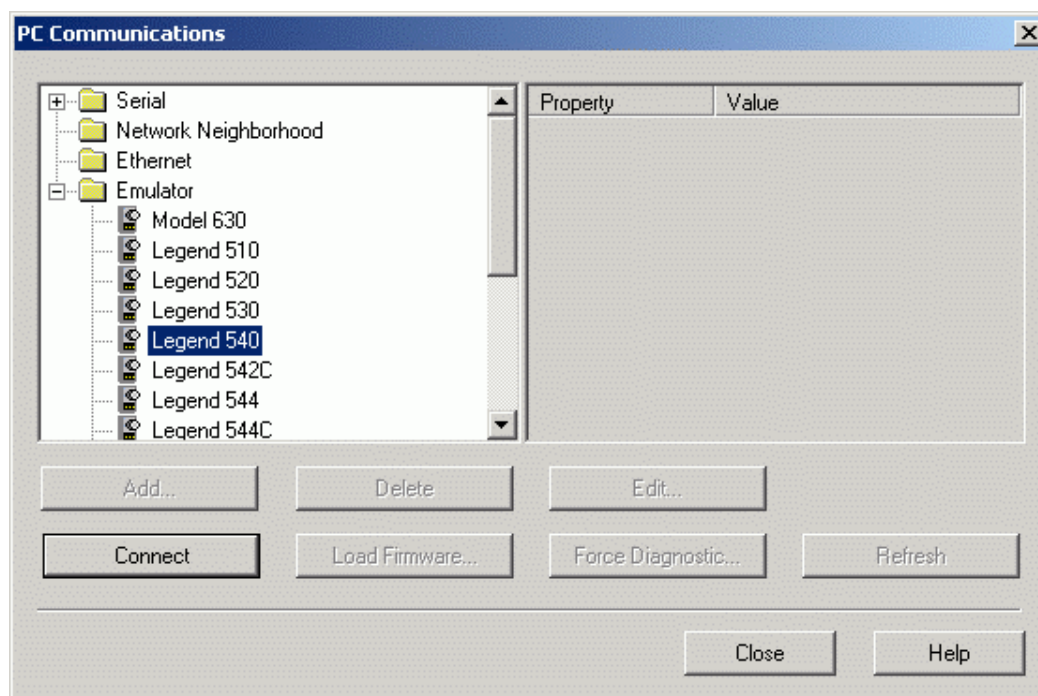
**Note:** Tags (data items) in the **InspectionParameters** and **InspectionStatistics** folders cannot be cut pasted, copied, deleted, or renamed.



#### Inspection Statistics Properties in the DVT OPC Configurator

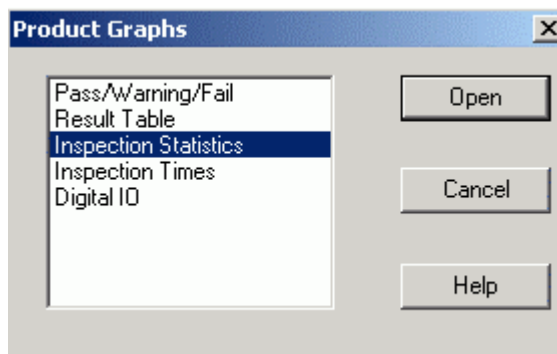
You can view inspection statistics using the DVT FrameWork by doing the following:

1. From the Windows **Start** menu, select **Programs > DVT Applications > FrameWork**. This opens the **FrameWork** application interface.
2. In the PC Communications dialog box, select a device (e.g. "Legend 540" Emulator) from the tree control, as shown in the figure below. Click the **Connect** button.



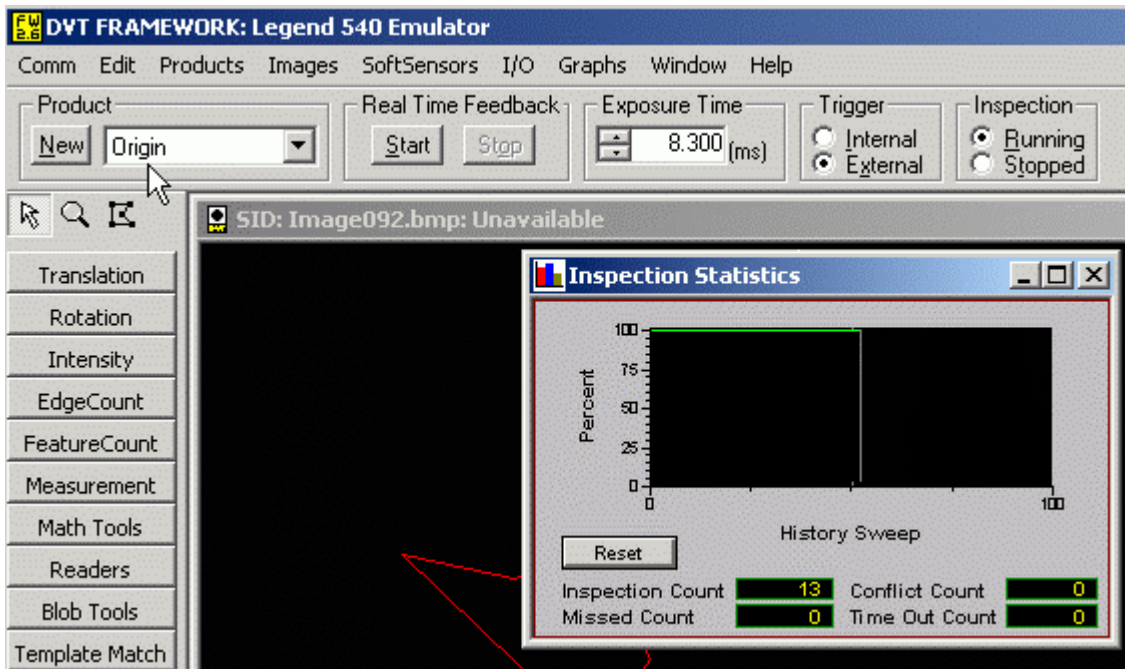
### Selecting a Device

3. In the DVT Framework user interface, select **Product Graphs** from the **Graphs** menu. Select **Inspection Statistics** from the **Product Graphs** dialog box, as shown in the figure below. Click **Open**.



### Selecting the "Inspection Statistics" Product Graph

4. The Inspection Statistics graph opens in the DVT Framework, as shown in the figure below. In the toolbar, select **Running** under **Inspection**, and select **External** under **Trigger**.



Viewing Inspection Statistics

## 7.4 Alarm and Event Logging Support

The SmartImage Sensors have the ability to log events to an Event Logging service on the TCP/IP network. When the IP address and a port number of an event-logging service are specified in the SmartImage Sensor, the event logging can be activated. When a SmartImage Sensor activates event logging, it creates a TCP/IP connection to the service and can start logging events. The OPC server acts as an Event Logging service in order to receive these events and expose them to OPC clients. The OPC Server does not log the events locally, but rather uses the OPC Alarms and Events mechanism to notify the OPC clients.

The default port number for event logging is port 50000. However, this value is configurable through the OPC Server configuration user interface.

## 7.5 TraceWorX Support

TraceWorX provides online diagnostics and tuning of the DVT OPC Server. TraceWorX is designed expressly for systems integrators, OEMs and customers who want to have tools for doing their own troubleshooting and diagnostics.

TraceWorX tracks the runtime activity for the DVT OPC Server and logs the runtime data to a log file based on user-configured trace levels. The log file provides a thorough, color-coded report detailing all activity for the application, including the time, the date, the severity level, and a description of the event or problem.

TraceWorX also features several options for reporting issues to technical support. If you are experiencing problems with any applications, the log file deployment options, such as compressing and e-mailing log files, are ideal for tracking and archiving data and sending detailed reports to technical support. Developers can use these reports to identify the source of the problems.

To start the TraceWorX Configuration Utility:

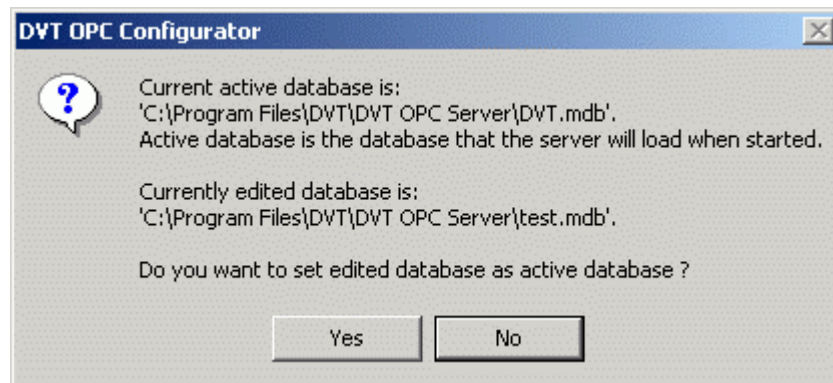
1. From the Windows **Start** menu, select **Programs > DVT Applications > DVT OPC Server > TraceWorX32**.
2. The **TraceWorX32.htm** file, which contains the Configuration Utility, opens in your Web browser.

**Note:** You can also open the **TraceWorX32.htm** file directly from the DVT OPC Server installation directory.

For help on using the TraceWorX Configuration Utility, please see the **TraceWorX32 Help.htm** file in the product installation by selecting **Start > Programs > DVT Applications > DVT OPC Server > TraceWorX32 Help**.

## 7.6 Activating the Database

Once your configuration is complete, you need to make sure that it is the active database. The database that is currently active is the one that the server uses. To make the current database active, select **Make Active** from the **File** menu. A dialog box appears showing both the current active database and database that is currently being edited, as shown in the figure below. To set the edited database as the active database, click the **Yes** button. Next time the server goes into runtime, it will use this active database for all of its operations.



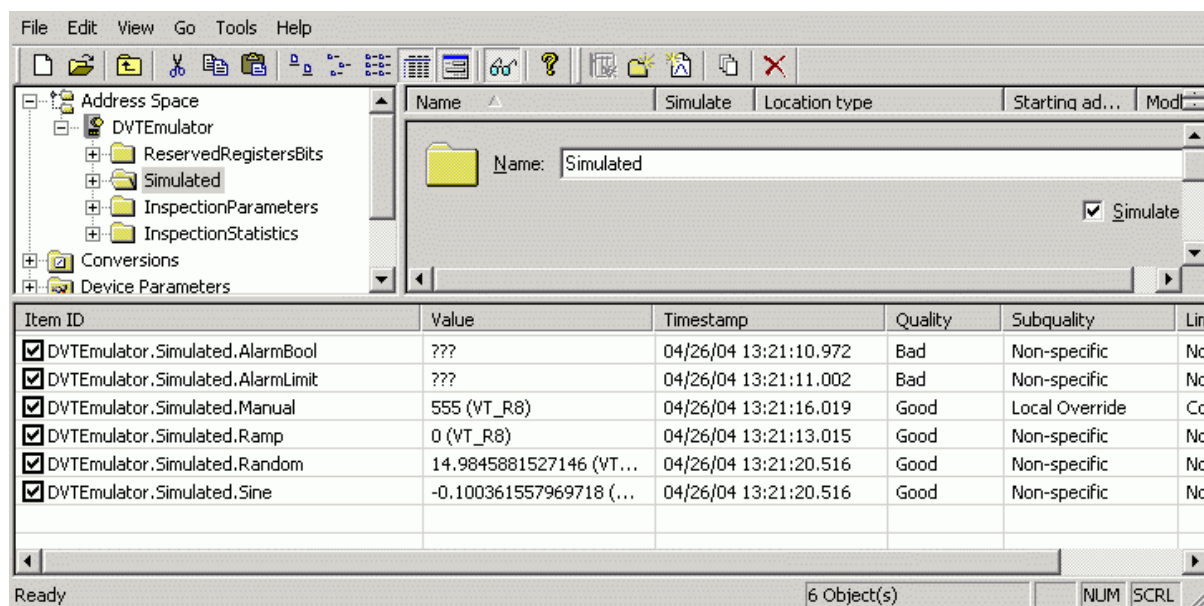
Activating the Database

## 7.7 Monitor View

The DVT OPC Configurator includes a runtime monitor for viewing server data. To change to the monitor view, select **Monitor View** from the **View** menu. The runtime monitor appears in the bottom pane of the Configurator screen, as shown in the figure below. During runtime, the monitor scans the server and displays the tag values and other statistics such as date, time, and quality.

**Note:** In order to view data in the monitor view, you must select a folder in the Configurator **Address Space** tree control that contains data items.

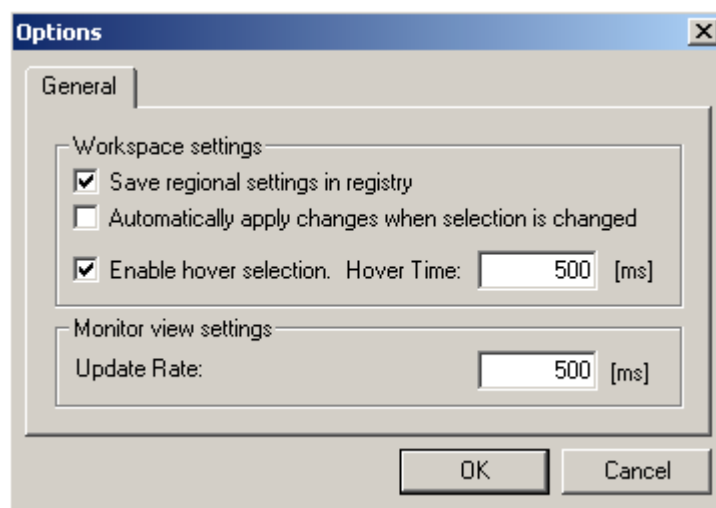
Monitoring is enabled for each item with a check mark next to it. To enable/disable monitoring for an item, you can click on the box to the left of the item. A check mark inside the box means the item is enabled for monitoring. If there is no check mark, then the item is disabled.



Runtime Monitor View

### 7.7.1 Setting the Monitor View Refresh Rate

To set the **OPC Monitoring Update Rate** (in milliseconds), select **Options** from the **Tools** menu and click on the **General** tab of the **Options** dialog box, as shown in the figure below.



Options Dialog Box: General Tab

### 7.7.2 Enable Monitoring

Monitoring is enabled for each item with a check mark next to it. To enable/disable monitoring for an item, you can click on the box to the left of the item. A check mark inside the box means the item is enabled for monitoring. If there is no check mark, then the item is disabled. To enable monitoring for an unchecked item in the monitor view, you can also right-click on the item and select **Enable Monitoring** from the pop-up menu, as shown in the figure below.

| Item ID                                                                     | Value                   | Timestamp             | Quality |
|-----------------------------------------------------------------------------|-------------------------|-----------------------|---------|
| <input type="checkbox"/> SmartImageEmulator.Simulated.AlarmBool             | NA                      |                       | NA      |
| <input checked="" type="checkbox"/> SmartImageEmulator.Simulated.AlarmLimit | ???                     | 03/24/04 10:45:18.287 | Bad     |
| <input checked="" type="checkbox"/> SmartImageEmulator.Simulated.Ramp       | 0 (VT_R8)               | 03/24/04 13:56:07.961 | Good    |
| <input checked="" type="checkbox"/> SmartImageEmulator.Simulated.Random     | 101.565599536           | 03/24/04 14:00:32.701 | Good    |
| <input checked="" type="checkbox"/> SmartImageEmulator.Simulated.Sine       | 0.958521321026127 (V... | 03/24/04 14:00:32.701 | Good    |

Enabling Monitoring for an Item

### 7.7.3 Disable Monitoring

Monitoring is enabled for each item with a check mark next to it. To enable/disable monitoring for an item, you can click on the box to the left of the item. A check mark inside the box means the item is enabled for monitoring. If there is no check mark, then the item is disabled. To disable monitoring for a checked item in the monitor view, you can also right-click on the item and select **Disable Monitoring** from the pop-up menu, as shown in the figure below.

| Item ID                                                                     | Value | Timestamp             | Quality |
|-----------------------------------------------------------------------------|-------|-----------------------|---------|
| <input checked="" type="checkbox"/> SmartImageEmulator.Simulated.AlarmBool  | ???   | 03/24/04 10:45:18.287 | Bad     |
| <input checked="" type="checkbox"/> SmartImageEmulator.Simulated.AlarmLimit |       | 03/24/04 10:45:18.287 | Bad     |
| <input checked="" type="checkbox"/> SmartImageEmulator.Simulated.Ramp       |       | 03/24/04 13:56:07.961 | Good    |
| <input checked="" type="checkbox"/> SmartImageEmulator.Simulated.Random     |       | 03/24/04 14:00:32.701 | Good    |
| <input checked="" type="checkbox"/> SmartImageEmulator.Simulated.Sine       |       | 03/24/04 14:00:32.701 | Good    |

Disabling Monitoring for an Item

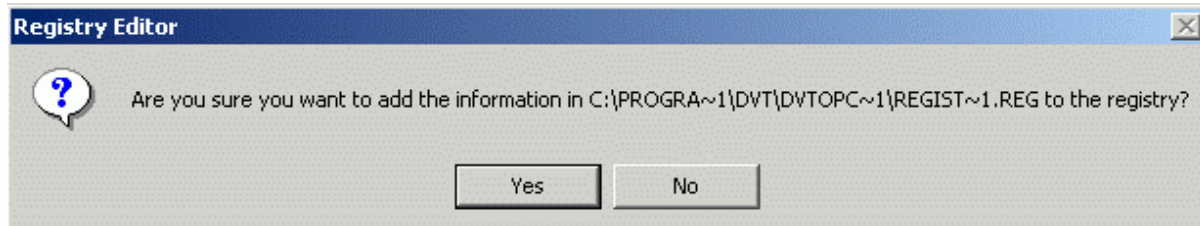
## 7.8 Registering the OPC Server as a Service

The DVT OPC Server can run as a service or an executable. The product installation automatically installs a **RegistrationContextMenu.reg** file, which enables you to register/unregister components by right-clicking on them.

To register the DVT OPC Server as a service:

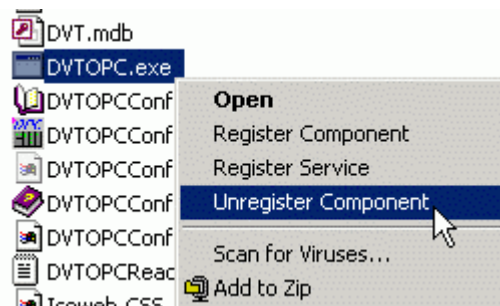
1. Go to the DVT OPC Server installation directory.

2. To initialize the **RegistrationContextMenu.reg** file, double-click on the file. You will get a message asking you if you want to add the file to the registry. Click **Yes**, as shown in the figure below.



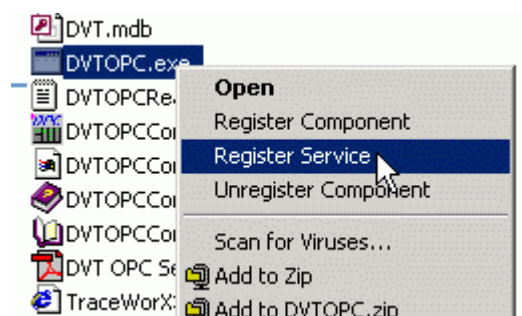
### Initializing the Registration File

3. Before registering the **DVTOPC.exe** file as a service, first unregister the file by right-clicking on it and selecting **Unregister Component** from the pop-up menu, as shown in the figure below.



### Unregistering the OPC Server Component

4. Right-click on the **DVTOPC.exe** file and select **Register Service** from the pop-up menu, as shown in the figure below.



### Registering the OPC Server as a Service