



ICONICS, Inc.
100 Foxborough Blvd.
Foxborough, MA 02035
Phone: (508) 543-8600
Fax: (508) 543-1503
[Worldwide Support Centers](http://www.iconics.com)
<http://www.iconics.com>



© ICONICS, Inc. All rights reserved.
Specifications are subject to change without notice.
GENESIS32, GENESIS64, Pocket GENESIS, BizViz
and their respective modules, OPC-To-The-Core,
and Visualize Your Enterprise are trademarks of
ICONICS, Inc. Other product and company names
mentioned herein may be trademarks of their
respective owners.

CONTENTS

[Introduction](#)

[ScriptWorX Architecture](#)

[Starting ScriptWorX](#)

[Creating a New Configuration Database](#)

[Toolbars](#)

[Menus](#)

[Options](#)

[Projects](#)

[Designers](#)

[Scripts](#)

[VBA Script Wizard](#)

[VBA Tutorial](#)

Introduction

ScriptWorX has undergone some large changes from ScriptWorX32 to ScriptWorX 2010. ScriptWorX32 is still included with GENESIS32 Version 9, so ScriptWorX32 configurations from earlier versions will not need to be updated, but the many advantages of ScriptWorX 2010 will most likely convince you to use it for new applications.

ScriptWorX 2010 allows you to create and run custom scripts. These scripts are written in Microsoft Visual Basic for Applications (VBA), and are run as multithreaded objects. The ScriptWorX 2010 Configurator indicates what type of script is being created and how it will be executed. The robust, stand-alone ScriptWorX 2010 application runs as an NT service. ScriptWorX 2010 allows the creation and management of Visual Basic for Applications (VBA) scripts, and its unique multi-tasking, multi-processor environment allows multiple scripts to run concurrently. User-defined VBA scripts can perform OPC read-and-write operations to any OPC server, which interfaces to factory floor devices, such as PLCs and other OPC-compliant servers. ScriptWorX 2010 simultaneously performs calculations, manages databases, and executes any operation available in the VBA language to access reports or perform recipe operations. ScriptWorX 2010 also offers project-level scripting as part of GENESIS32.

ScriptWorX 2010 also takes advantage of the new Unified Data Manager with its extensive centralized support for triggers. VBA scripts can be scheduled and configured for execution once or repeated daily, weekly, or monthly, and perform reliably on a 24 x 7 x 365 yearly schedule with a simple point-and-click interface. Scripts can be triggered based on OPC Data Access, OPC Alarm and Events, and file events.

Features of ScriptWorX 2010

Key features of ScriptWorX 2010 include:

- Configurator based on familiar, standardized configuration toolkit
- Run as an NT service
- New threading model
- Predictable multithreading
- Robustness including watchdogs and auto-recovery
- Variables and status as OPC data tags
- Trigger information exposed to scripts
- Integrated Microsoft Visual Basic Editor
- Monitor natively in MonitorWorX
- Global triggers via Unified Data Manager and Unified Data Browser

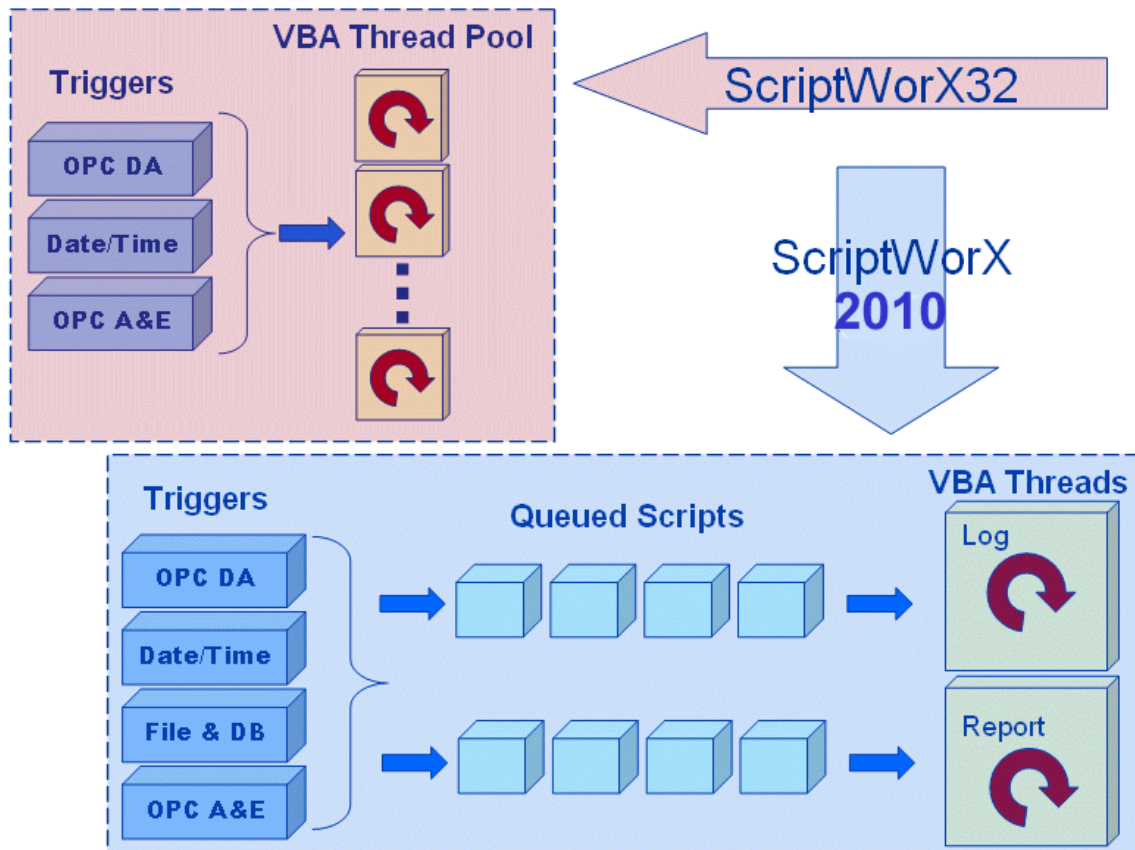
ScriptWorX Components

The ScriptWorX 2010 application consists of three main components:

- **ScriptWorX User Interface (Configurator):** The ScriptWorX Configurator is where you select trigger conditions for scripts. On start it automatically launches the ScriptWorX Engine.
- **ScriptWorX Server (ScriptWorX Engine):** The ScriptWorX Engine is a server (invisible to users) that communicates between the ScriptWorX user interface and the Visual Basic Editor. It creates scripts based on the configuration in the user interface.
- **Integrated Microsoft Visual Basic Editor:** The greatest advantage of VBA is the availability of multithreading. VBA allows for creating, editing, and debugging multithreaded VBA scripts.

ScriptWorX Architecture

The figure below illustrates the architectural differences between the legacy ScriptWorX32 (thread pool) and the new ScriptWorX 2010 (designated thread queue).



ScriptWorX Architecture

Predictable Multithreading

In the legacy ScriptWorX32, each time a script is triggered it is executed in a new thread. The maximum number of threads allowed was specified in the settings, and if no threads were available the scripts would queue up and wait for a thread to become free. This could sometimes cause scripts to execute in an unpredictable order.

In ScriptWorX 2010, scripts are organized into VBA designers, and each designer runs in its own thread. The number of designers in your project specifies the number of threads available. If multiple scripts are triggered within a thread, the scripts will queue up within the thread. The limit of queuing scripts is specified on a per-designer basis, as is the behavior when this limit is exceeded. This new threading model gives the user much more control over the threads used by ScriptWorX 2010. This is a huge advantage over ScriptWorX32.

Features of ScriptWorX 2010 predictable multithreading include:

- Share components by scripts in the same thread (e.g. database connection).
- Share Visual Basic for Applications (VBA) variables by scripts in the same thread.
- **OnTimer** event per thread; scripts can execute per millisecond within the thread.
- **OnValueChange** event per thread; scripts execute when a global variable is modified.
- Scripts are triggered by universal triggers.
- Scripts are queued in designated thread queue.
- Scripts are executed by VBA runtime threads.
- Inter-thread synchronization by using global variables and other native sync objects.

Robustness and Optimization

ScriptWorX 2010 watchdogs:

- Allow detecting and auto-restarting dead threads.
- Require cooperation from script.

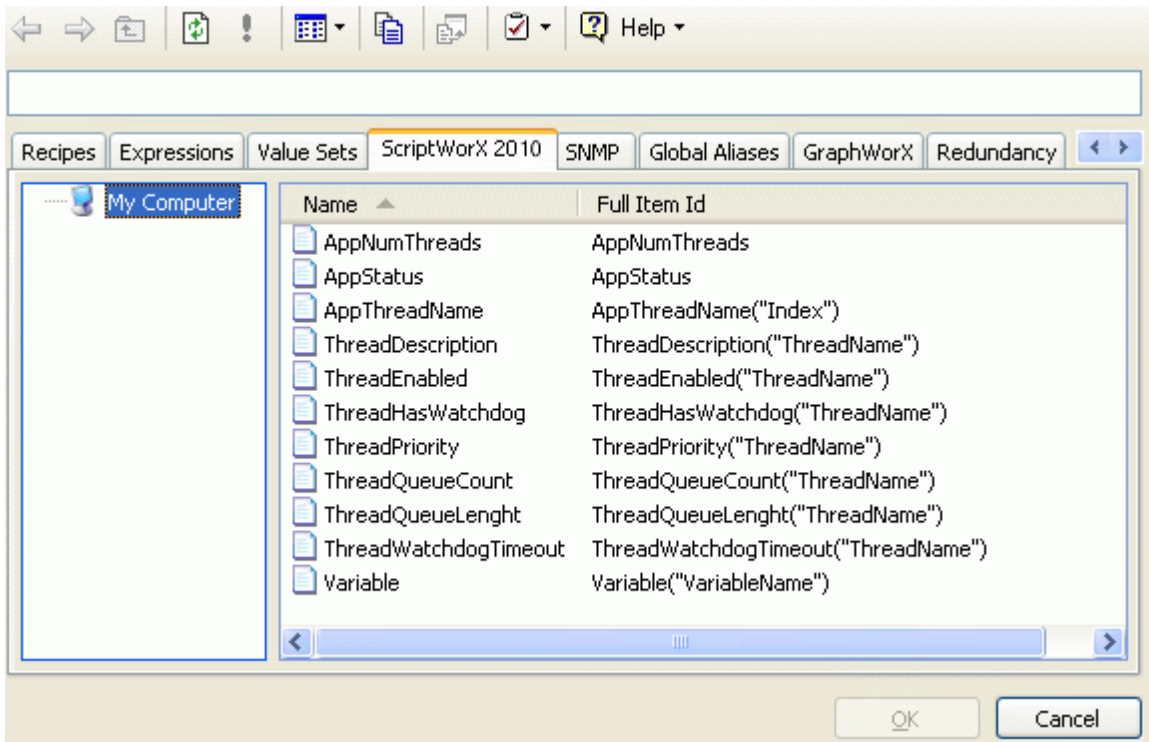
ScriptWorX 2010 optimization includes:

- Thread priorities.
- Thread startup squelch time.
- What to do when script queue is full: Discard new scripts, discard old scripts, or define longer queue.

OPC Connectivity

ScriptWorX 2010 has much more built-in functionality when dealing with OPC Data Access (DA) tags.

- VBA global variables are exposed as **Data Tags**, which can be used similarly to OPC DA tags in ICONICS applications via the GenBroker plug-in. Data tags can also be available as "true" OPC DA tags via DataWorX32. ScriptWorX's data tags can be seen in the **ScriptWorX 2010** tab in the Unified Data Browser, as shown in the figure below.
- The status of a thread is also exposed as OPC DA tags. OPC clients can access the length of the queue, the remaining time in the Watchdog, how many times a script was executed, etc.
- ScriptWorX 2010 has built-in functions to read OPC DA tags.



ScriptWorX 2010 Tab in Unified Data Browser

Thread status is exposed as OPC tags:

- Queue length
- Number of executions
- Watchdog remaining time

Built-in OPC read and write functions include:

- `myval = Opc.Read( tag, timeout )`
- `Call Opc.Write( tag, myval, timout )`

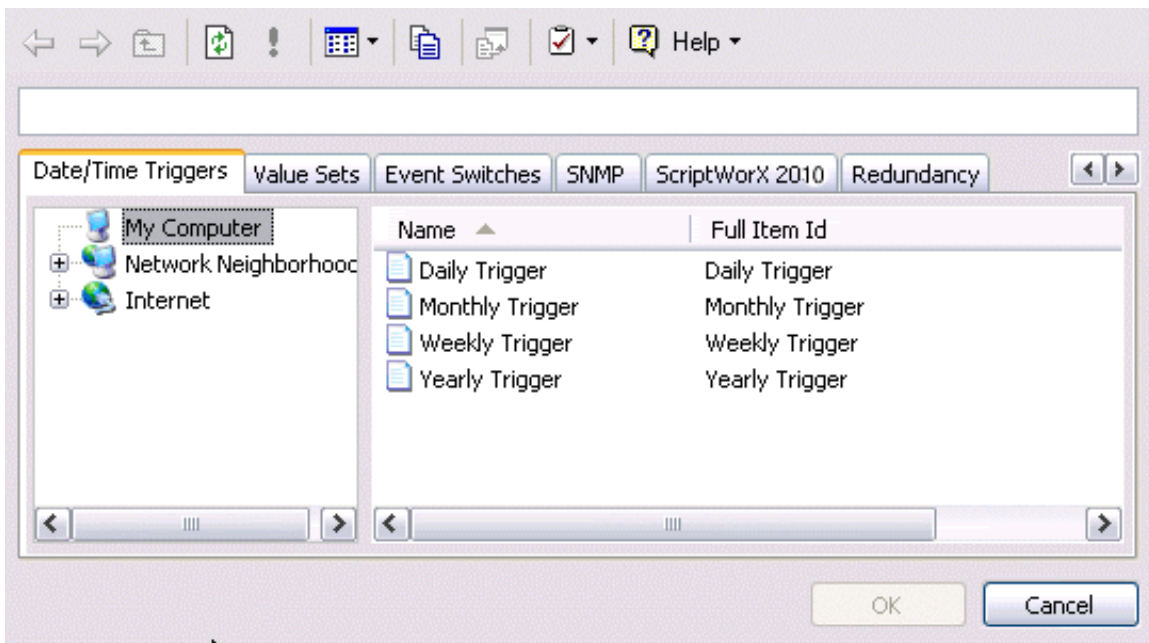
ScriptWorX Configurator

The **ScriptWorX Configurator**, which is described in detail in the following sections, provides a hierarchical configuration interface for your scripts:

- **Projects:** Projects are associated with .vba files that contain all of the code for the project.
- **Designers:** Each project may contain one or more **Designers/Threads**. The .vba file for a project is split into multiple designers, each of which is associated with its own thread in runtime. Any functions, subroutines and global variables within a designer are available to all scripts in that designer.
- **Scripts:** Scripts link a particular subroutine within the designer to a trigger created with the Unified Data Manager. Whenever the trigger happens for a script, the associated subroutine will enter the queue for the Designer/Thread.

Script Triggers

Script triggers are configured in the Unified Data Manager as global triggers. In the script configuration dialog within the ScriptWorX Configurator, you can launch the Unified Data Browser, which allows you to conveniently select global triggers to associate with scripts, as shown in the figure below. For more information about the Unified Data Manager, please see the Unified Data Manager help documentation.



Selecting Triggers From the Unified Data Browser

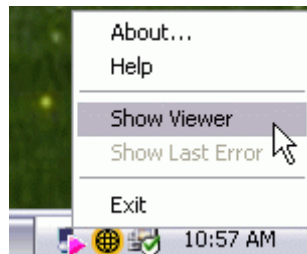
Integrated Microsoft Visual Basic for Applications

The Microsoft **VBA Editor** can be launched directly from the script configuration dialog within the ScriptWorX 2010 Configurator. Using the VBA Editor, you can create, edit, and debug multithreaded VBA scripts. For more information, please refer to the Microsoft VBA help documentation.

Runtime Monitor

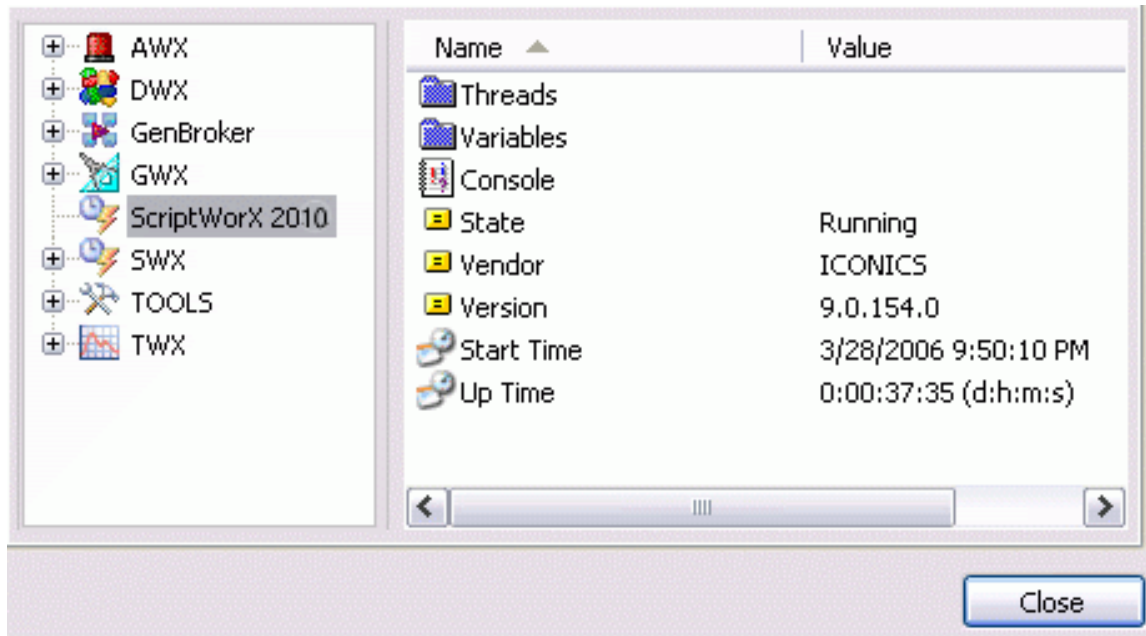
ScriptWorX monitoring is now implemented in **MonitorWorX**:

1. Launch MonitorWorX from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > MonitorWorX**.
2. To display the MonitorWorX user interface, right-click on the MonitorWorX icon in the Windows taskbar and select **Show Viewer** from the pop-up menu, as shown in the figure below.



Displaying the MonitorWorX Viewer

3. In the tree control, select **ScriptWorX 2010**, as shown in the figure below. This displays monitoring information for the following:
 - Application status
 - Runtime console
 - Thread details and thread console
 - Global variable Information



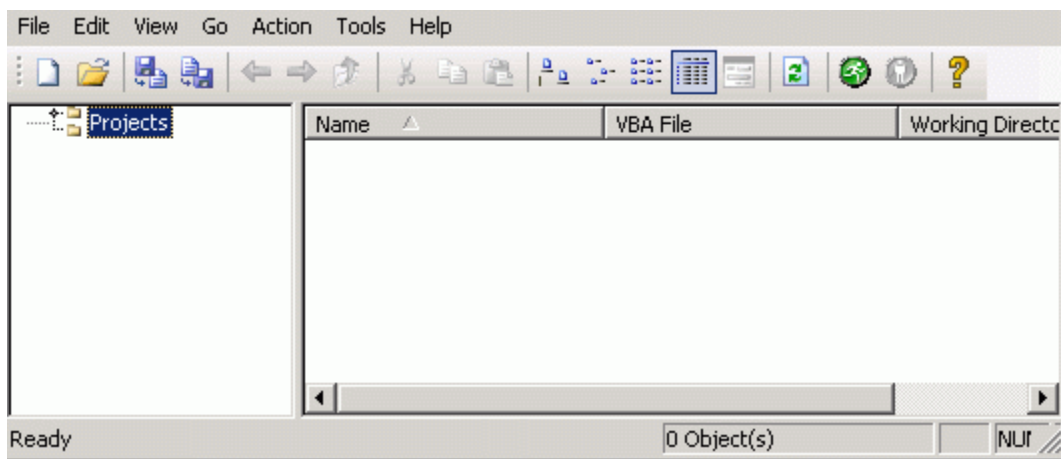
ScriptWorX Monitoring in MonitorWorX

For more information about MonitorWorX, please see the MonitorWorX help documentation.

Starting ScriptWorX

To start the ScriptWorX 2010 Configurator:

1. From the Windows **Start** menu, select **Programs > ICONICS GENESIS32 > ScriptWorX > ScriptWorX 2010**.
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.

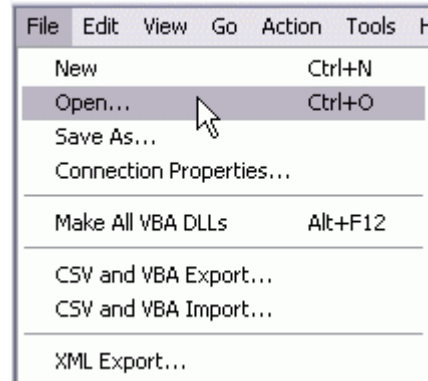


Configurator Screen

Creating a New Configuration Database

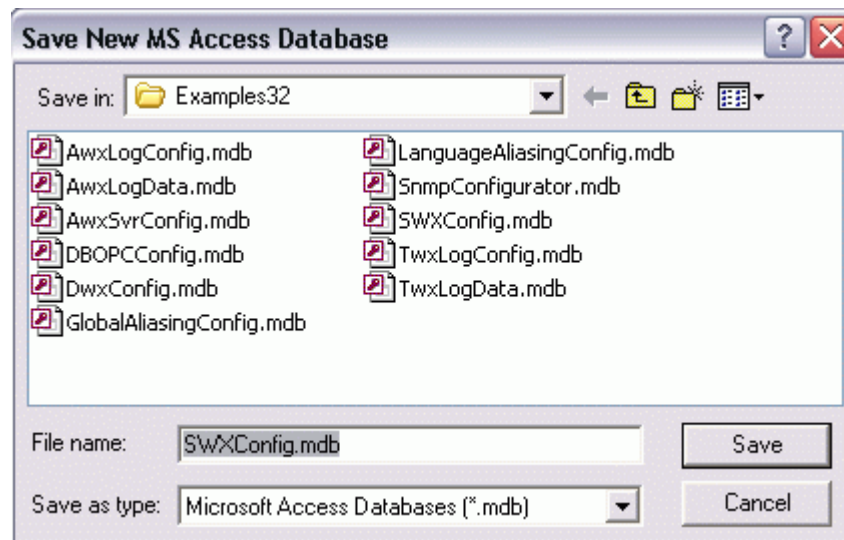
To create a new configuration database in the Configurator:

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



Creating a Configuration Database

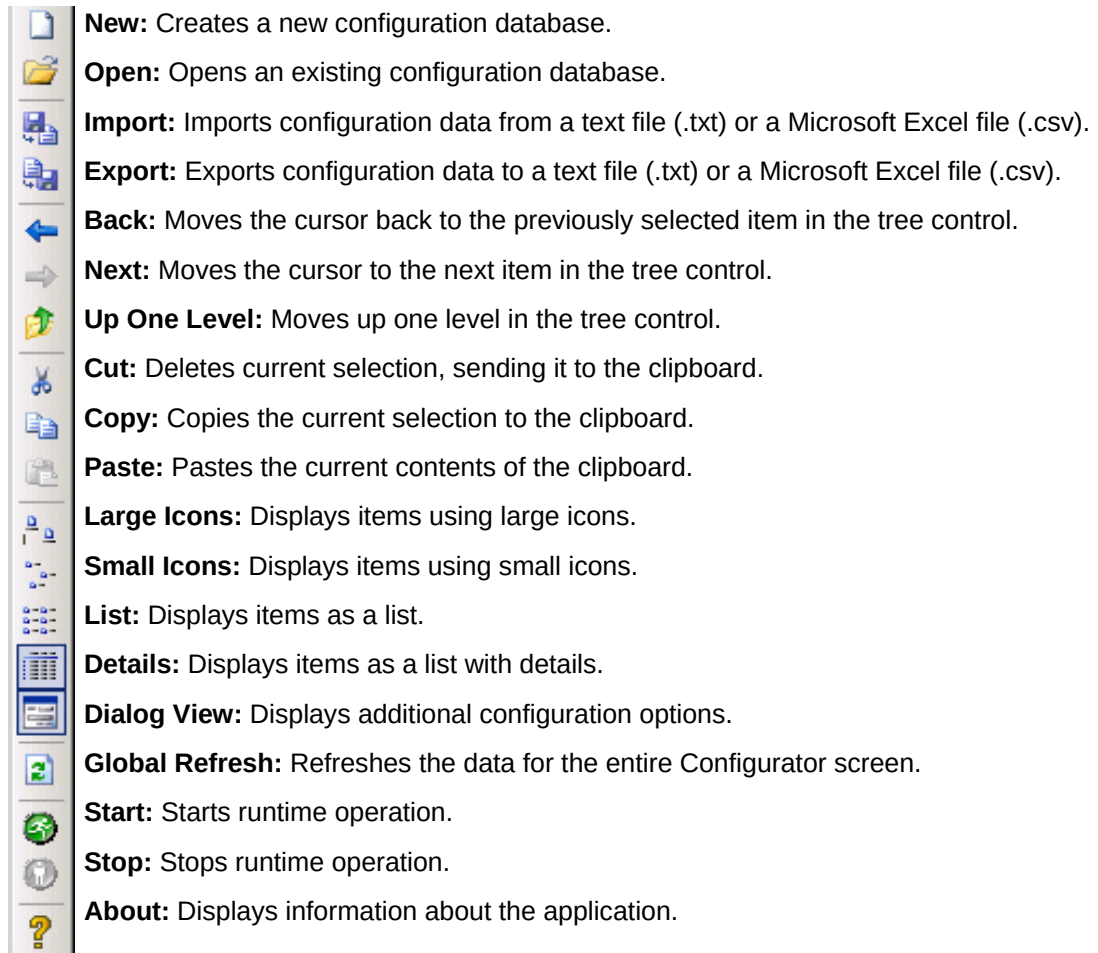
2. In the **Save New Database** dialog box, shown in the figure below, browse for the target directory, give the file a name, and then click the **Save** button.



Saving the New Configuration Database

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.



Menus

The menu bar of the Configurator contains the following menus:

[File](#)

[Edit](#)

[View](#)

[Go](#)

[Action](#)

[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.

File Menu

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

Command	Shortcut Key	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 as a Microsoft Access (.mdb) file.
Connection Properties		Displays the current database connection properties.
Make All VBA DLLs	ALT+F12	Allows you to build the .dll files without actually being in VBA. This is helpful since it is not always desired to build the .dll directly after entering the code; you may wish to change the configuration of the script (i.e. what triggers it) after you have written the script.
CSV and VBA Export		Packs the CSV configuration file together with all .vba files used in the configuration into a .cab file (common Microsoft Cabinet file).
CSV and VBA Import		Imports configuration data from a cabinet .cab file. The .cab file is unpacked to one .csv file and several .vba files.

<u>XML Export</u>		Exports configuration data to an XML file.
<u>XML Export Schema</u>		Exports configuration data to an XML Schema file.
<u>XML Import</u>		Imports configuration data from an XML file.
<u>XML Validate</u>		This feature does not import an XML data file, but it will try to validate its structure using stored XML schema. Once it passes this validation, the XML file is acceptable for import by the Configurator.
<u>Make Active</u>		Activates the current database. If this command is not available, then the current database is already the active database.
<u>Exit</u>		Closes the application.

File-New

Selecting **New** from the **File** menu creates a new configuration database.

File-Open

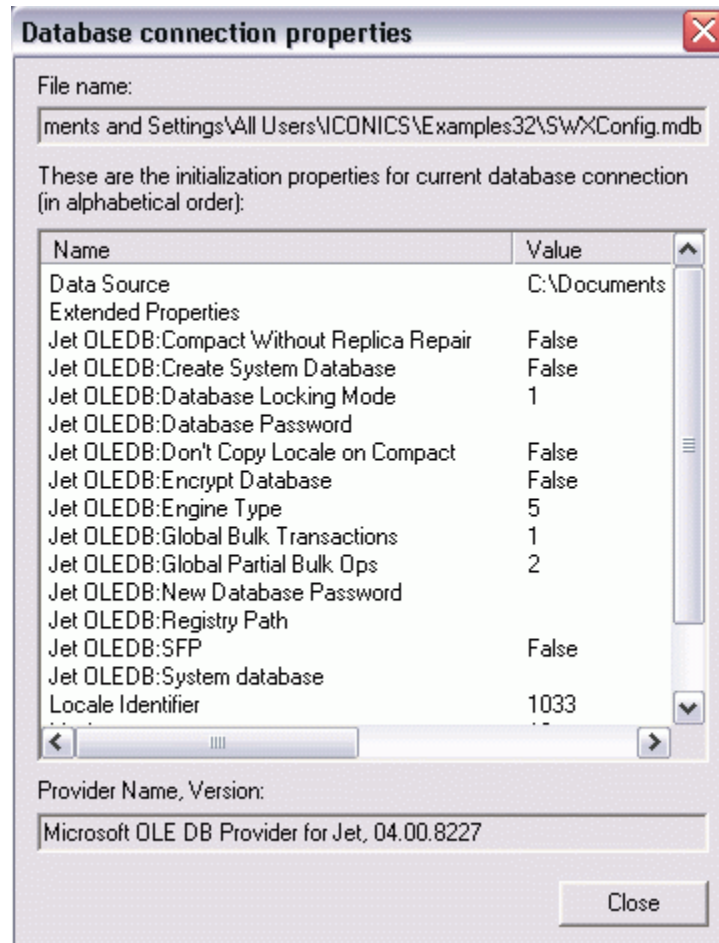
Selecting **Open** from the **File** menu opens an existing configuration database.

File-Save As

Selecting **Save As** from the **File** menu saves the current database under a different name as a Microsoft Access (.mdb) file.

File-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

File-Make All VBA DLLs

Selecting **Make All VBA DLLs** from the **File** menu allows you to build the .dll files without actually being in VBA. This is helpful since it is not always desired to build the .dll directly after entering the code; you may wish to change the configuration of the script (i.e. what triggers it) after you have written the script.

File-Export CSV and VBA

Selecting **CSV Export** from the **File** menu packs the CSV configuration file together with all .vba files used in the configuration into a .cab file (common Microsoft Cabinet file).

File-Import CSV and VBA

Selecting **CSV Import** from the **File** menu imports configuration data from a cabinet .cab file. The .cab file is unpacked to one .csv file and several .vba files.

File-XML Export

Selecting **XML Export** from the **File** menu exports configuration data to an XML file.

File-XML Export Schema

Selecting **XML Export Schema** from the **File** menu exports configuration data to an XML schema file.

File-XML Import

Selecting **XML Import** from the **File** menu imports configuration data from an XML file.

File-XML Validate

Selecting **XML Validate** from the **File** menu does not import an XML data file, but it will try to validate its structure using stored XML schema. Once it passes this validation, the XML file is acceptable for import by the Configurator.

File-Make Active

Selecting **Make Active** from the **File** menu activates the currently edited database. Next time the server goes into runtime, it will use this active database for all of its operations.

File-Recent

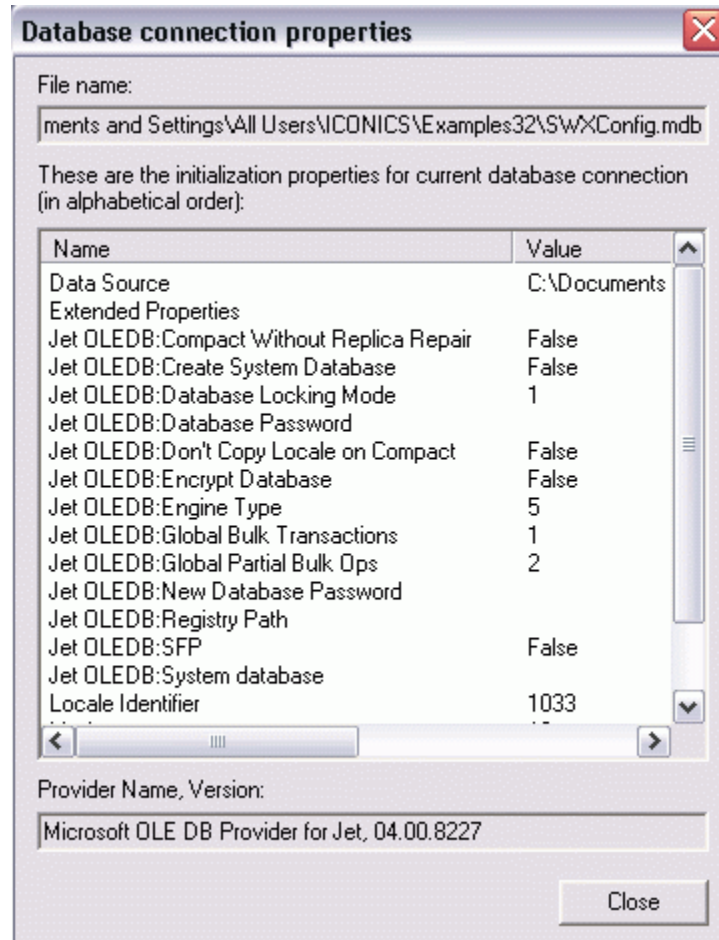
The **File** menu lists the most recently used configuration databases. To open a recently used database, simply select the name of the database from the **File** menu.

File-Exit

Selecting **Exit** from the **File** menu closes the application. The current configuration database is automatically saved.

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.



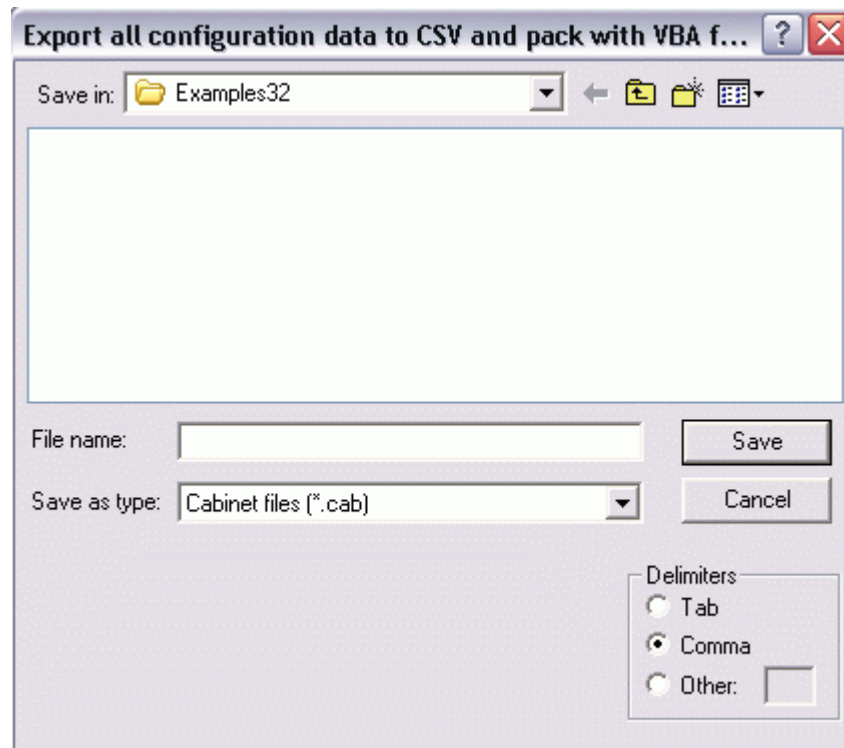
Database Connection Properties Dialog Box

Exporting Configuration Data

Exporting Data to a CSV File and Packing With VBA Files

The export/import functionality in the ScriptWorX 2010 Configurator is different from that of other GENESIS32 configurators. ScriptWorX 2010 cannot simply export the configuration database into a CSV or TXT file, because it does not work without VBA files. Each project created in the configuration has its .vba file that contains the VBA code with designers/threads and scripts/procedures. So there is a different export/import functionality based on the packing of the CSV configuration file together with all .vba files used in the configuration into a .cab file (common Microsoft Cabinet file).

To export the configuration data, select **CSV and VBA Export** from the **File** menu. This opens the **Export Configuration Data** 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** 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.



Exporting Configuration Data

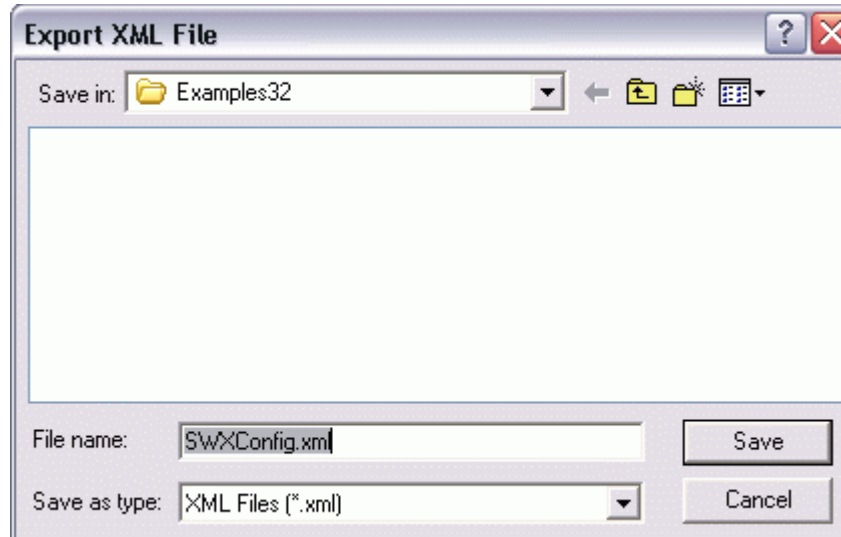
Exporting Data to an XML File

The Configurator also allows you to export data from your configuration database to an XML file. The XML export/import functionality was mainly developed for Windows platforms that do not support databases (e.g. Windows CE and Windows Embedded). XML has the following advantages over the CSV import/export function:

- XML has a standardized format, unlike the text/CSV format, which uses various delimiters (e.g. TAB instead of commas, strings could not accept all characters, etc.)
- XML is language-independent, whereas CSV converts date/time, floats, and currency fields according to local settings in Windows. For example, using CSV, you cannot export data on German Windows and import it on English windows without making changes
- Windows has an installed automation object that has the capability to work with XML. Thus, programmers can create/modify their configurations outside the Configurator using Visual Basic, if desired.
- XML supports schemas. A **schema** is a special XML file that specifies the data structure of an XML data file.

To export data, select **XML Export** from the **File** menu. This opens the **Export XML File** dialog box, as shown in the figure below. Give the file a name, and then choose the directory to which you want to export the data from your database. Click **Save**.

Note: You can also export configuration data to an XML Scheme file by selecting **XML Export** from the **File** menu.



Exporting Configuration Data to an XML File

Importing Configuration Data

Importing and Unpacking Configuration Data and VBA Files

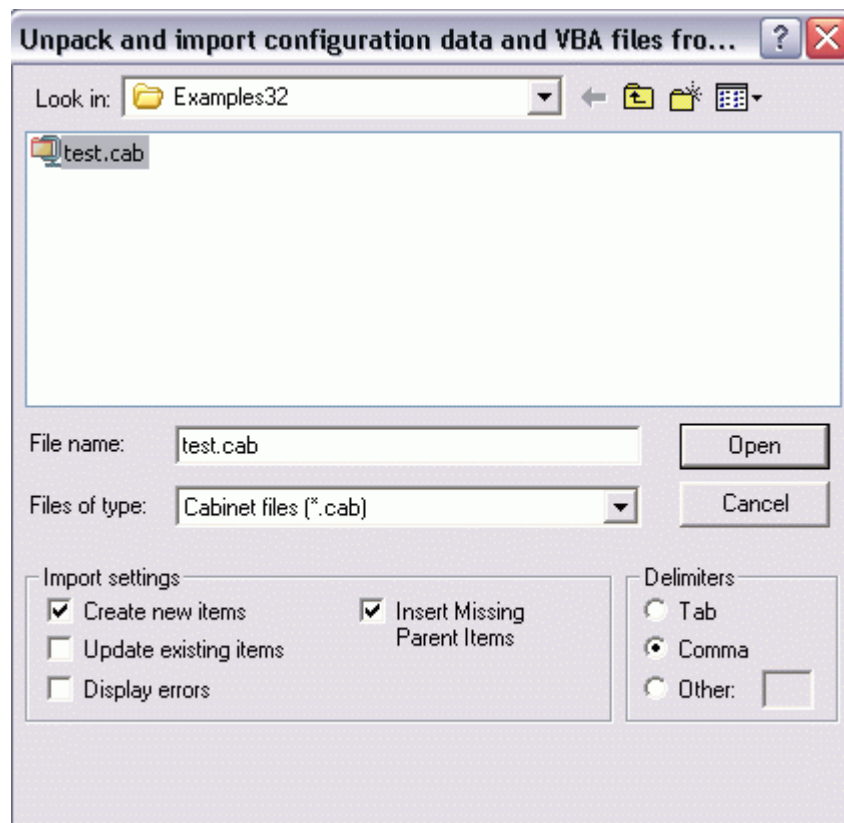
When importing data in ScriptWorX 2010, the configuration is imported from a cabinet .cab file. The .cab file is unpacked to one .csv file and several .vba files. The .csv file is imported into the configuration database via the common CSV import feature, and all the .vba files are copied into their locations (according to the project configuration). Configuration data are always exported/imported into/from the .cab file, and the configuration database always into/from the .csv file.

To import data, select **CSV and VBA Import** from the **File** menu. This opens the **Import Configuration Data** 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.

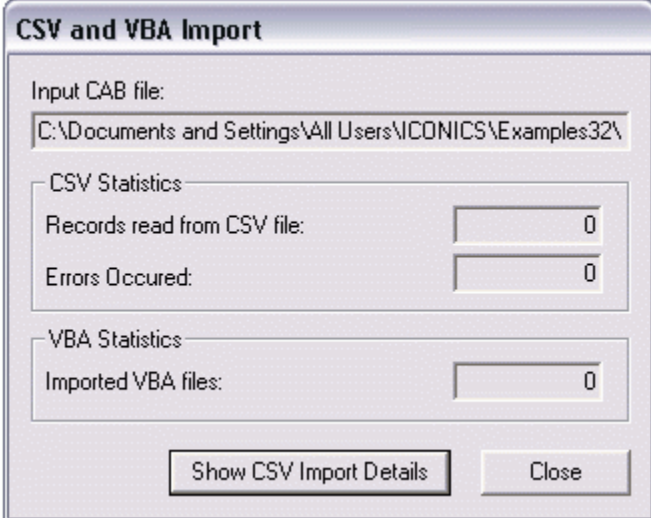
Note: Either **Create new items** or **Update existing items** must be selected. Otherwise there is nothing to import.

- **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.



Importing Configuration Data

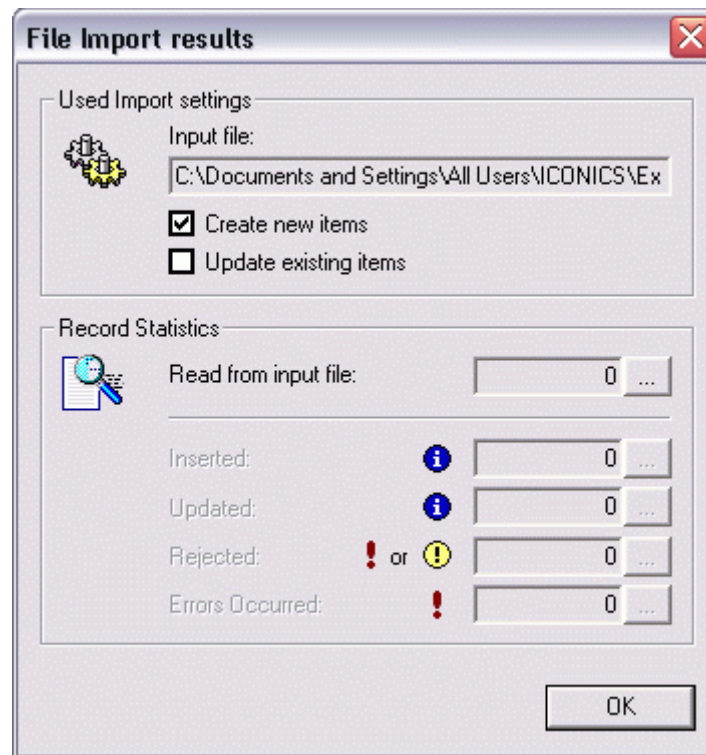
When you have selected a file to import, click **Open**. The **CSV and VBA Import** dialog box, shown in the figure below, summarizes the number of records imported from the CSV file and the number of VBA imported files.



The image shows a Windows-style dialog box titled "CSV and VBA Import". It has a light purple border and a white background. At the top, there is a label "Input CAB file:" followed by a text box containing the path "C:\Documents and Settings\All Users\ICONICS\Examples32\". Below this, there are two sections: "CSV Statistics" and "VBA Statistics". The "CSV Statistics" section contains two labels: "Records read from CSV file:" and "Errors Occured:", each followed by a text box containing the number "0". The "VBA Statistics" section contains one label: "Imported VBA files:", followed by a text box containing the number "0". At the bottom of the dialog box, there are two buttons: "Show CSV Import Details" and "Close".

CSV and VBA Import Results Dialog Box

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



CSV File Import Results Dialog Box

Importing Data From an XML File

The Configurator allows you to import data from your configuration database to an XML file. The XML export/import functionality was mainly developed for Windows platforms that do not support databases (e.g. Windows CE and Windows Embedded). XML has the following advantages over the CSV import/export function:

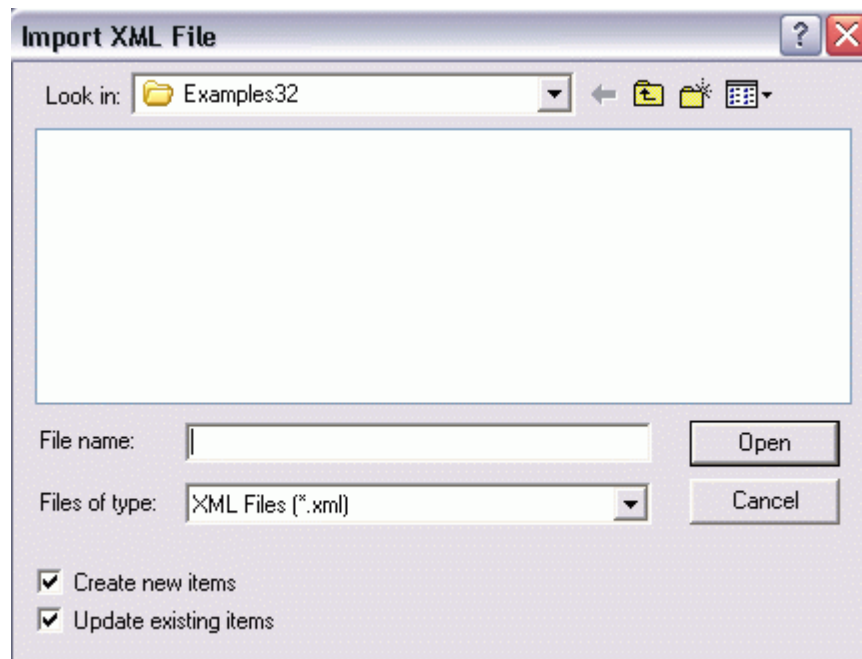
- XML has a standardized format, unlike the text/CSV format, which uses various delimiters (e.g. TAB instead of commas, strings could not accept all characters, etc.)
- XML is language-independent, whereas CSV converts date/time, floats, and currency fields according to local settings in Windows. For example, using CSV, you cannot export data on German Windows and import it on English windows without making changes
- Windows has an installed automation object that has the capability to work with XML. Thus, programmers can create/modify their configurations outside the Configurator using Visual Basic, if desired.
- XML supports schemas. A **schema** is a special XML file that specifies the data structure of an XML data file.

To import data, select **XML Import** from the **File** menu. This opens the **Import XML File** dialog box, as shown in the figure below. Give the file a name, and then choose the directory from which you want to import the data. You can then specify the delimiters and choose from the following import settings. Click **Open**.

- **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.

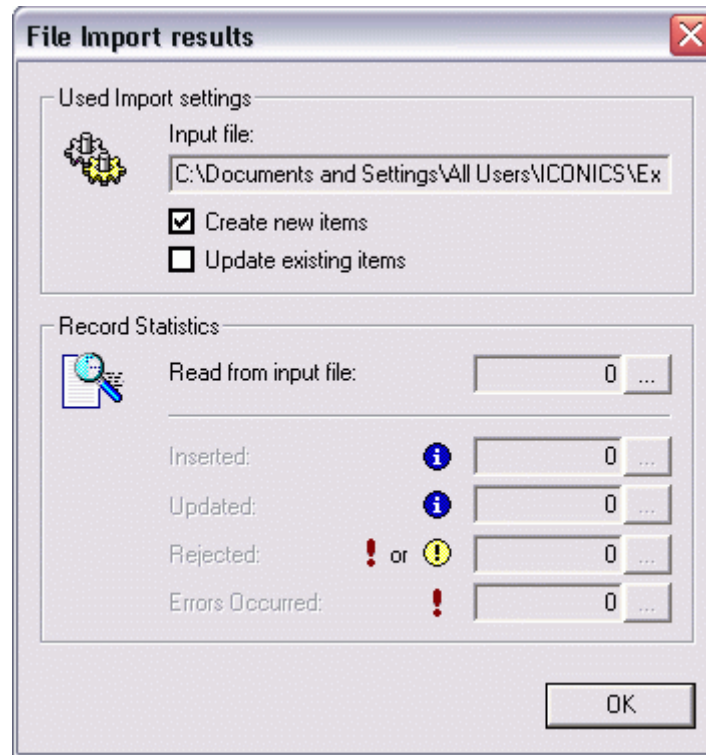
Note: Selecting **XML Validate** from the **File** menu does not import an XML data file, but it will try to validate its structure using stored XML schema. Once it passes this validation, the XML file is acceptable for import by the Configurator.



Importing Configuration Data From an XML File

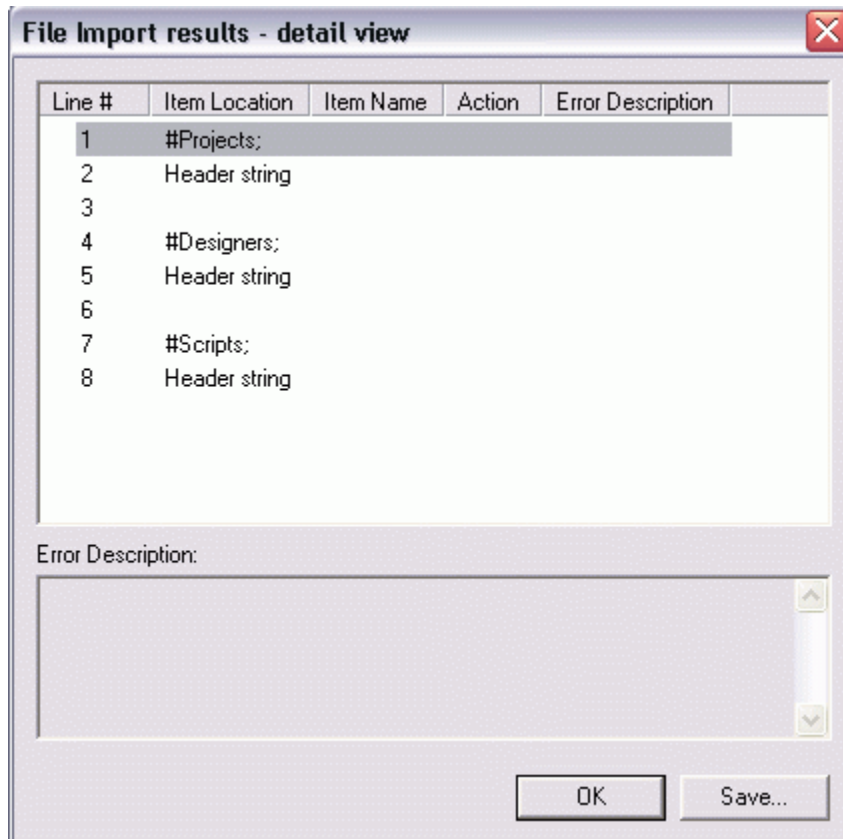
File Import Results

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.

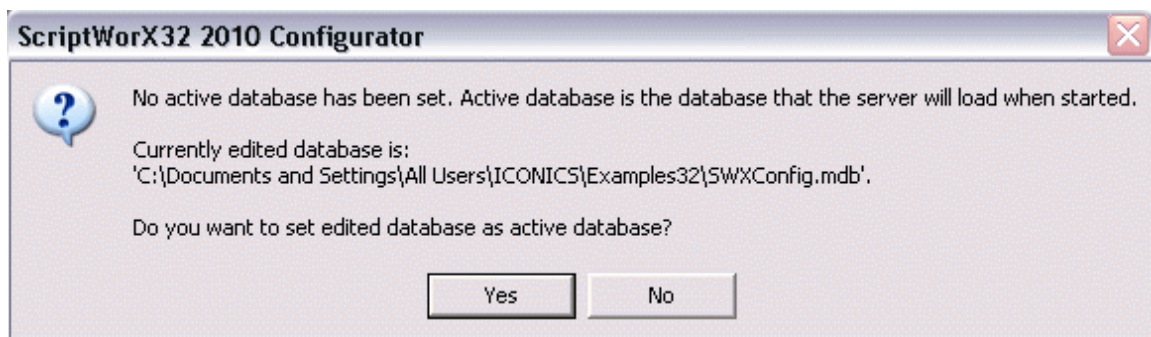


File Import Results Details

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. If the **Make Active...** selection is grayed out, then the current database is already the active database.

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.



Activating the Database

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 Multiply Item dialog box, which allows you to multiply an item in the tree control.
Find and Replace	CTRL+H	Opens the Find and Replace dialog box. Enter a string to search for in the Find What field. The Replace operation works in a similar way and replaces the Find What string(s) with the Replace With string(s). All dialog options are the same, except that the Replace dialog box has a field called Replace With .
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 in a list in the upper-right-hand section of the viewer.

Edit-New

Selecting **New** from the **Edit** menu creates a new item depending on what is selected in the tree control.

Edit-Multiply

Selecting **Multiply** from the **Edit** menu opens the **Multiply Item** dialog box, which allows you to duplicate the selected item in the tree control.

Edit-Find and Replace

Selecting **Find and Replace** from the **Edit** menu opens the **Find and Replace** dialog box. Enter a string to search for in the **Find What** field. The **Replace** operation works in a similar way and replaces the **Find What** string(s) with the **Replace With** string(s). All dialog options are the same, except that the **Replace** dialog box has a field called **Replace With**.

Edit-Cut

Selecting **Cut** from the **Edit** menu cuts the selected object from the view and places it on the clipboard.

Edit-Copy

Selecting **Copy** from the **Edit** menu copies the selected object to the clipboard.

Edit-Paste

Selecting **Paste** from the **Edit** menu pastes the last object placed on the clipboard.

Edit-Select All

Selecting **Select All** from the **Edit** menu selects all objects in a list. The selection is shown in the upper-right-hand section of the viewer.

Edit-Invert Selection

Selecting **Invert Selection** from the **Edit** menu unselects all selected items and selects all unselected items.

Edit-Delete

Selecting **Delete** from the **Edit** menu deletes the selected object.

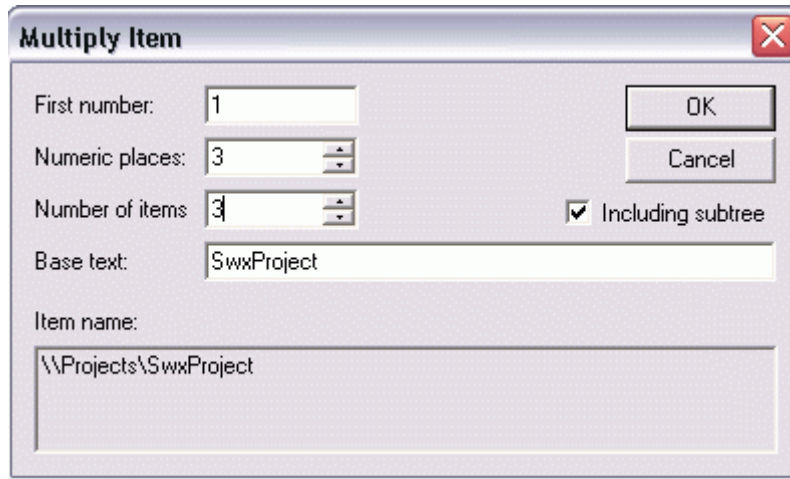
Edit-Rename

Selecting **Rename** from the tree control context menu allows you to change the name of the selected object.

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.
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 below.



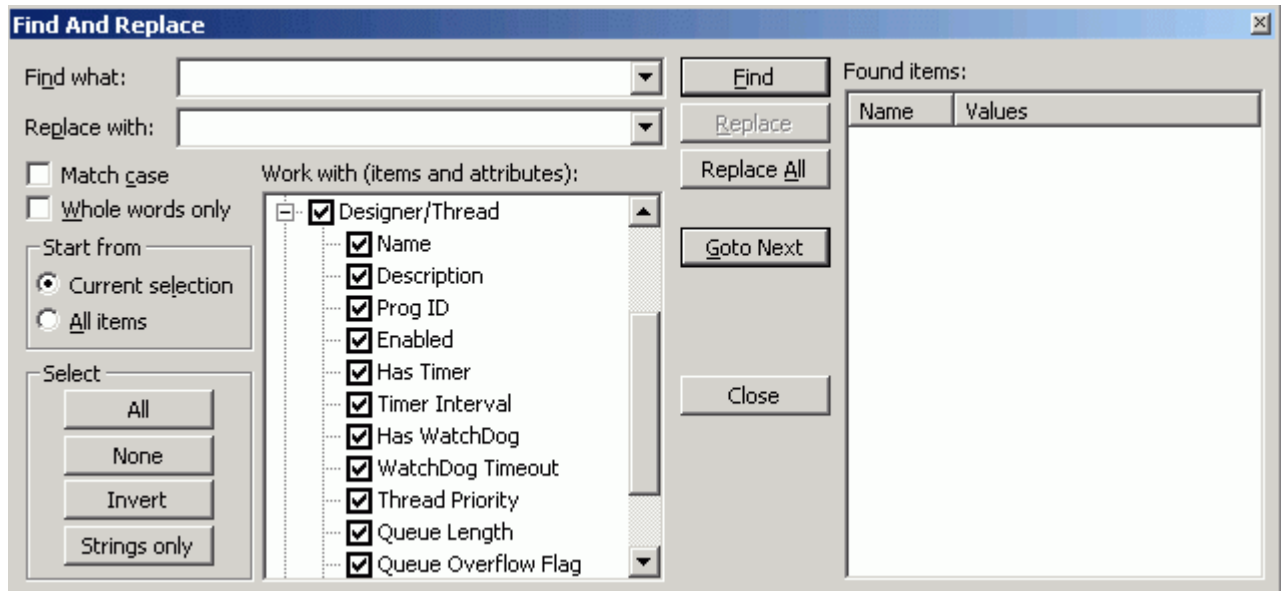
Multiply Item Dialog Box

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:
 - SWXProject001
 - SWXProject002
 - SWXProject003

All subfolders will also be multiplied.

Find and Replace

The **Find and Replace** function on the **Edit** menu allow you to find and replace strings. Selecting **Find and Replace** from the **Edit** menu opens the **Find and Replace** dialog box, as shown in the figure below. Enter a string to search for in the **Find What** field. The replace operation works in a similar way and replaces the **Find What** string(s) with the **Replace With** string(s). When the search or replace operation is completed, the results are displayed in the **Found Items** pane at the right of the **Find and Replace** dialog box



Find/Replace Dialog Box

In the **Work With (Items and Attributes)** field, check one or more types of properties to search for:

- **All:** Selects items and attributes.
- **None:** Unselects all items and attributes.
- **Invert:** Unselects the currently selected items and attributes, and selects the currently unselected items and attributes.
- **Strings Only:** Selects items and attributes that contain strings.

In the **Start From** field, select the scope of the search:

- **Current Selection:** Searches only the current selection.
- **All items:** Searches all items in the Configurator.
- **Current Display:** Searches the entire currently loaded display.
- **Multiple Files:** Searches in multiple display files. Click the ... button to open the **Edit File List** dialog box, which allows you to specify a list of files to search.

You can check one or both of the following search options:

- **Match whole word only:** Restricts the search operation to whole-word matches.
- **Match case:** Restricts the search operation to case-sensitive matches.

Find and Replace Operations

When you are ready to execute a find or replace operation, click one of the following buttons:

- **Go to Next:** Finds the next match.
- **Find:** Finds all occurrences of the search text.
- **Replace:** Replaces a single match and then advances to the next match.
- **Replace All:** Replaces all occurrences of the search text with the replacement text.
- **Close:** Stops an in-progress search or replace operation.

View Menu

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

View Menu Commands

Command	Shortcut Keys	Function
Toolbars		Toggles the toolbars.
Status Bar		Toggles the status bar.
Large Icons	F7	Displays items using large icons.
Small Icons	F8	Displays items using 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).
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 Select Language dialog box (see below). Choose the language you wish to use for your system (Unicode version only) and click OK . For navigation purposes, use the buttons and check boxes in the List section.
Global Refresh	F5	Refreshes the data for the entire Configurator screen.

Subtree Refresh	CTRL+F 5	Refreshes only the data contained in the currently selected subtree.
---------------------------------	-------------	--

View-Status Bar

Selecting **Status Bar** from the **View** menu toggles the status bar at the bottom of the Configurator window.

View-Large Icons

Selecting **Large Icons** from the **View** menu displays items using large icons.

View-Small Icons

Selecting **Small Icons** from the **View** menu displays items using small icons.

View-List

Selecting **List** from the **View** menu displays items as a list.

View-Details

Selecting **Details** from the **View** menu displays items as a list along with detailed information about the configuration of each item.

View-Sort By

Selecting **Sort By** from the **View** menu 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.

View-Show/Hide Columns

Selecting **Show/Hide Columns** from the **View** menu 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.

View-Global Refresh

Selecting **Global Refresh** from the **View** menu refreshes the data for the entire Configurator screen.

View-Subtree Refresh

Selecting **Subtree Refresh** from the **View** menu refreshes only the data contained in the currently selected subtree.

View-Select Language

Choosing **Select Language** from the **View** menu opens the **Select Language** dialog box, which allows you to choose which language to use in your display.

View-Dialog View

Selecting **Dialog View** from the **View** menu toggles the configuration window (right-hand pane).

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 English is checked, the languages will appear as their English name. If Localized is checked, the languages will appear with the native country in parentheses (for languages with several dialects only). When Native 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.

Go Menu

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

Go Menu Commands

Command	Shortcut Keys	Function
Back	CTRL+ALT + Left Arrow	Moves the cursor back to the previously selected item in the tree control.
Forward	CTRL+ALT + Right Arrow	Moves the cursor forward to the previously selected item in the tree control.
Up One Level		Moves the cursor up one level in the tree control.
Next Item	ALT+Down Arrow	Moves the cursor to the next item down in the tree control.
Previous Item	ALT+Up Arrow	Moves the cursor to the next item up in the tree view.
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.
Previous Pane	SHIFT+F6	Moves the cursor to the last pane used.

Go-Back

Selecting **Back** from the **Go** menu moves the cursor back to the previously selected item in the tree control.

Go-Forward

Selecting **Forward** from the **Go** menu moves the cursor forward to the previously selected item in the tree control.

Go-Up One Level

Selecting **Up One Level** from the **Go** menu moves the cursor up one level in the tree control.

Go-Next Item

Selecting **Next Item** from the **Go** menu moves the cursor to the next visible item in the tree control.

Go-Previous Item

Selecting **Previous Item** from the **Go** menu moves the cursor to the previous visible item in the tree control.

Go-Expand Item

Selecting **Expand Item** from the **Go** menu expands an item that contains a submenu.

Go-Collapse Item

Selecting **Collapse Item** from the **Go** menu collapses an item that contains a submenu.

Go-Page Up

Selecting **Page Up** from the **Go** menu moves the cursor up to the first item in the tree.

Go-Page Down

Selecting **Page Down** from the **Go** menu moves the cursor down to the last visible item in the tree.

Go-Home

Selecting **Home** from the **Go** menu moves the cursor up to the first item in the tree.

Go-End

Selecting **End** from the **Go** menu moves the cursor down to the last visible item in the tree.

Go-Next Pane

Selecting **Next Pane** from the **Go** menu moves the cursor to the next pane in the window.

Go-Previous Pane

Selecting **Previous Pane** from the **Go** menu moves the cursor to the last pane used.

Action Menu

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

Action Menu Commands

Command	Function
<u>Start Runtime</u>	Enters the display into runtime mode.
<u>Stop Runtime</u>	Stops runtime operation.

Action-Start Runtime

Selecting **Start Runtime** from the **Action** enters the display into runtime mode.

Action-Stop Runtime

Selecting **Stop Runtime** from the **Action** menu stops runtime operation.

Tools Menu

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

Tools Menu Commands

Command	Shortcut Keys	Function
Macros > Visual Basic Editor	ALT+F11	Shows/hides the Visual Basic Editor.
Set VBA Projects Folder		Specifies a directory for storing your Visual Basic for Applications projects.
Options		Launches the Options dialog box.
Compact/Repair MS Access Database		Opens the Compact/Repair MS Access Database dialog box.

Tools-Options

Selecting **Options** from the **Tools** menu opens the [Options](#) dialog box.

Tools-Compacting and Repairing Microsoft Access Databases

Selecting Compact/Repair MS Access Database from the Tools menu opens the [Compacting and Repairing Microsoft Access Databases](#) dialog box.

Tools-Macros

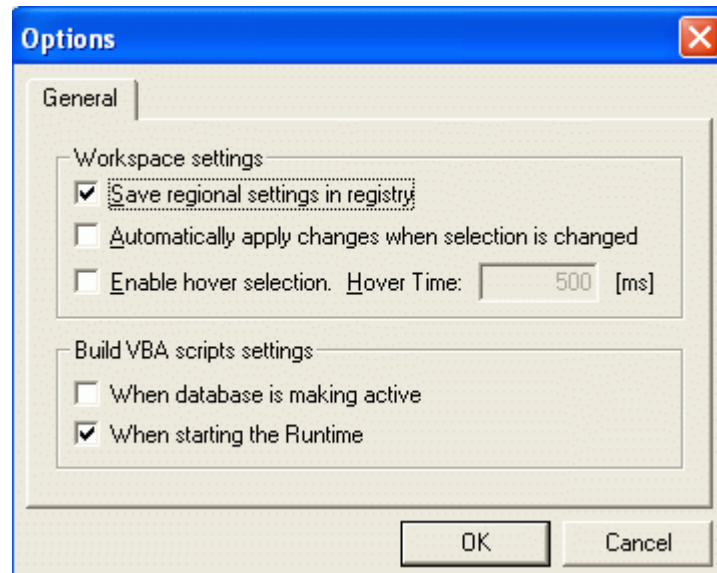
Selecting **Macros > Visual Basic Editor** from the **Tools** menu shows/hides the Visual Basic Editor.

Tools-Set VBA Projects Folder

Selecting **Set VBA Projects Folder** from the **Tools** menu opens the [Set VBA Projects Folder](#) dialog box, which allows you to specify a directory for storing your Visual Basic for Applications projects.

Options

To choose additional settings, select **Options** from the **Tools** menu. This opens the **Options** dialog box, as shown in the figure below. The **General** tab of the **Options** dialog box contains the following options:



Options Dialog Box: General Tab

Workspace Settings

- **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).

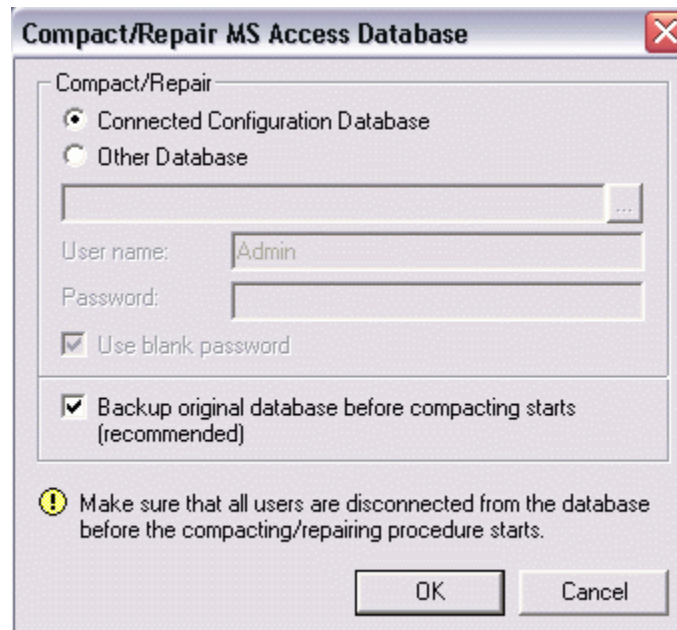
Build VBA Scripts Settings

- **When database is making active:** When this check box is checked, the configurator will automatically rebuild all the VBA files when the configuration database is activated.
- **When starting the runtime:** When this check box is checked, the configurator will automatically rebuild all the VBA files when the configuration database when runtime is started.

Compacting and Repairing Microsoft Access Databases

You can compact Microsoft Access databases, which can be either configuration databases or historical databases, using the **Compact/Repair MS Access Database** dialog box, shown in the figure below. To open this dialog box, select **Compact/Repair MS Access Database** from the **Tools** menu. Microsoft Access-based databases are subject to database fragmentation over time, and the support for the database will compact the target database, reclaim unused space, and drastically improve database performance.

Note: It is critical that no users or client applications are connected to the database at the time of compacting and that, if the **Backup Original Database** option is selected, there is plenty of available hard disk space.



Compacting and Repairing Microsoft Access Databases

Selecting a VBA Projects Folder

To specify a directory for your VBA projects, select **Set VBA Projects Folder** from the **Tools** menu. Click the **Browse** button to select a directory, as shown in the figure below.

Note: If no specific folder for VBA projects is set, then the folder where the configuration database is saved is also used for the VBA projects.



Specifying a VBA Projects Folder

Help Menu

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

Command	Shortcut Key	Function
Help Topics	F1	Launches the online Help for the Configurator.
About Application		Launches the About Box , which contains information about the product version number, copyright, and available disk space. It also contains contact information.

Help-Help Topics

Selecting **Help Topics** from the **Help** menu launches the online Help for the Configurator. Press the F1 button for context-sensitive help topics.

Help-About Application

Selecting **About Application** from the **Help** menu launches the **About Box**, which contains information about the product version number and copyright.

Projects

Projects are associated with .vba files that contain all of the code for the project.

Creating a New Project

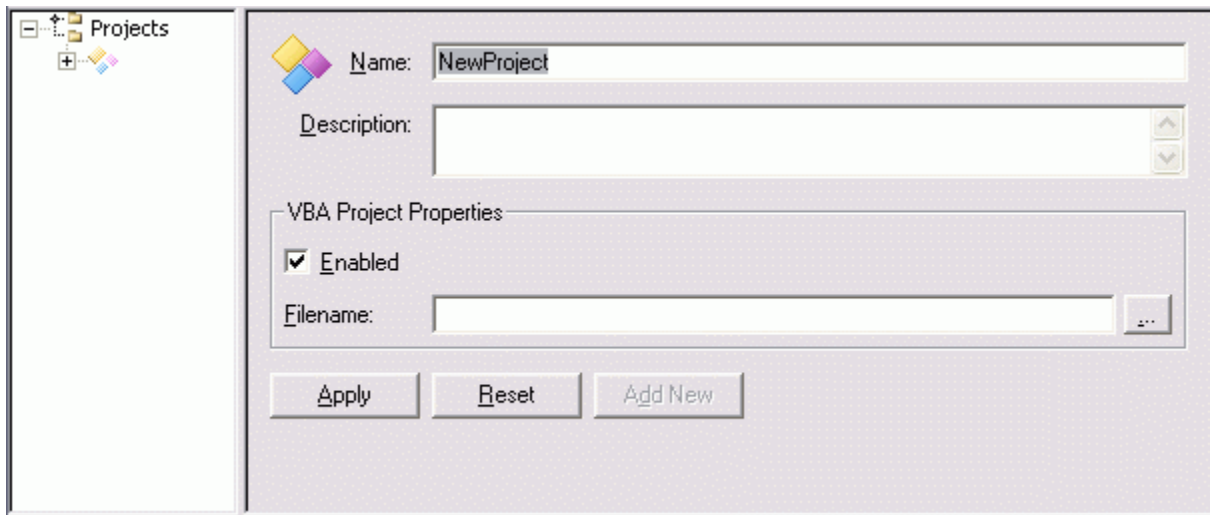
To create a new ScriptWorX 2010 project:

1. Right-click on the **Projects** tree control of the Configurator and select **New > Project** from the pop-up menu, as shown in the figure below.



Creating a New Project

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



Setting the Properties for the New Project

3. In the **Name** field, type a name for the new project.
4. In the **Description** field, you can type an optional description for the new project.
5. When you have finished configuring the project properties, click the **Apply** button. The new project appears under the **Projects** tree control.

Project Properties

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

- **Name:** Enter a name for the project.
- **Description:** Enter a description for the project (optional).
- **Enabled:** Activates the VBA script project. A script project must be enabled in order to be run.
- **File Name:** Specifies the name of the .vba project file, which contains all the code for the project.
- **Apply:** Saves all changes to the configuration database.
- **Reset:** Clears all recently added information.
- **Add New:** Creates a new project to be configured.

The screenshot shows the 'Project Properties' dialog box. On the left, there is a 'Projects' pane with a tree view containing one project icon. The main area of the dialog is divided into sections. The top section has a 'Name' field with the text 'NewProject' and a 'Description' field which is empty. Below these is a section titled 'VBA Project Properties' which contains a checked checkbox labeled 'Enabled' and a 'Filename' field which is empty. At the bottom of the dialog are three buttons: 'Apply', 'Reset', and 'Add New'.

Project Properties

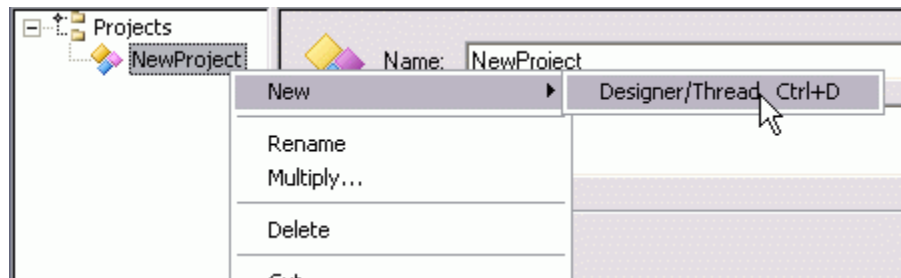
Designers

Within a project are **Designer/Threads**. The .vba file for a project is split into multiple designers, each of which is associated with its own thread in runtime. Any functions, subroutines and global variables within a designer are available to all scripts in that designer.

Creating a New Designer

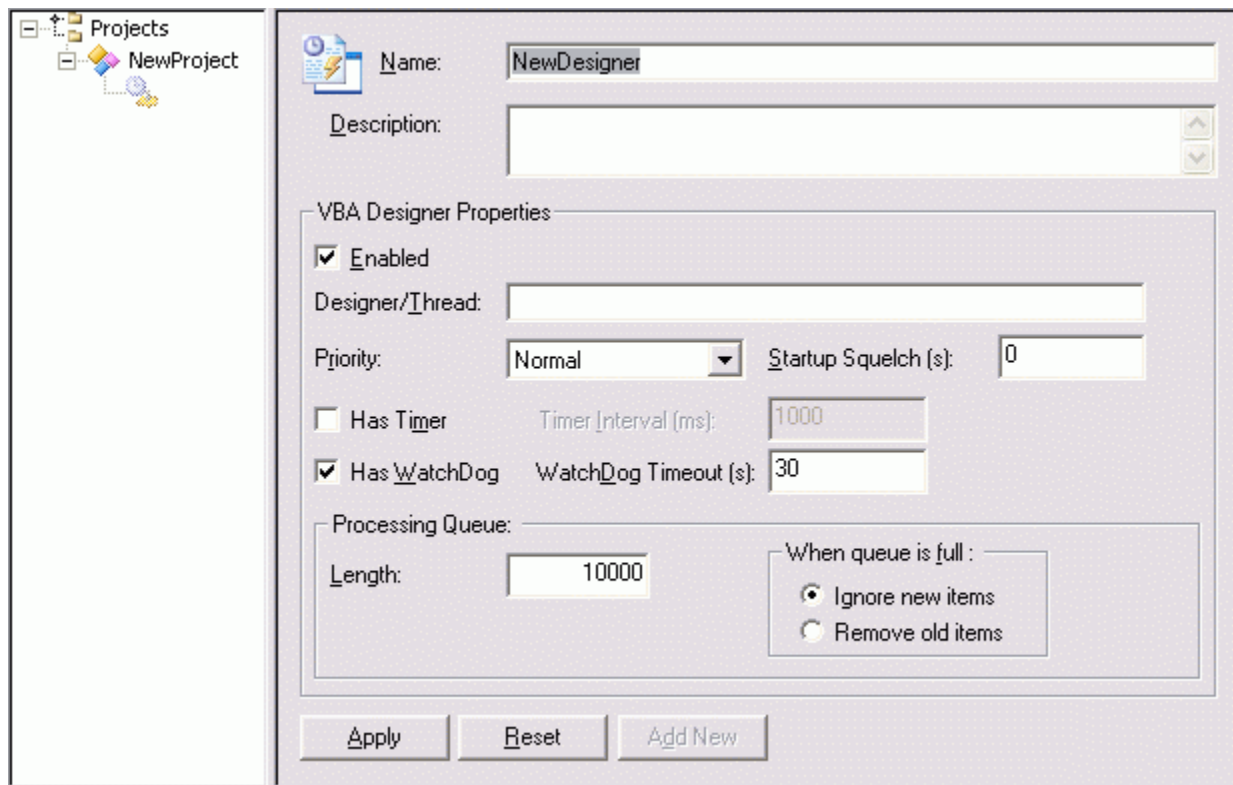
To create a new designer/thread:

1. Right-click on a project in the tree control of the Configurator and select **New > Designer/Thread** from the pop-up menu, as shown in the figure below.



Creating a New Designer/Thread

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



Setting the Properties for the New Designer

3. In the **Name** field, type a name for the new designer.
4. In the **Description** field, you can type an optional description for the new designer.
5. When you have finished configuring designer/thread properties, click the **Apply** button. The new designer/thread appears under the selected project in the tree control.

Designer Properties

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

- **Name:** Enter a name for the designer.
- **Description:** Enter a description for the designer (optional).
- **Enabled:** Activates the VBA designer.
- **Designer/Thread:** Enter a name for the VBA designer.
- **Priority:** Each designer/thread can be given one of seven priorities (Real-time, High, Above Normal, Normal, Below Normal, Low, and Idle), allowing the you to better manage when the threads are given processor time.
- **Startup Squelch:** Specifies the amount of time (in seconds) that the thread execution will be suppressed. The thread will be prevented from executing the specified number of seconds after ScriptWorX 2010 is started. A squelch of zero indicates the thread may start as soon as it is able.
- **Has Timer:** A timer can be created for each designer/thread. This timer behaves similar to the GENESIS32 Timer ActiveX. It can be used as a simple way to time events within a script without re-triggering the entire script.
- **Timer Interval:** Specifies the interval (in milliseconds) for the timer.
- **Has Watchdog:** Enabling the watchdog allows for detecting and auto-restarting dead threads. The watchdog prevents scripts from running out of control. If a watchdog is enabled for a designer/thread, any scripts that run longer than the specified number of seconds will be killed. Individual scripts can override the watchdog if needed.
- **Watchdog Timeout:** Specifies default the period of time (in seconds) before the watchdog timeout occurs.
- **Processing Queue Length:** Specifies the length of the specified thread queue for scripts. If multiple scripts within a designer/thread are triggered to run at the same time, they will queue up. The queue can be configured to hold only a certain amount of scripts and to either ignore new scripts or drop the old ones when that limit is reached.
- **When queue is full:** Specifies what to do when the script queue is full: Discard new scripts (ignore new items), discard old scripts (remove old items), or define a longer queue (length).
- **Apply:** Saves all changes to the configuration database.
- **Reset:** Clears all recently added information.
- **Add New:** Creates a new designer/thread to be configured.

Name	Enabled	VBA Script	Trigger Name
 LogToWord	Yes	LogToWord	trg_data:RandomTimeSine



Name: WordLoggerDesigner

Description:

VBA Designer Properties

☒ Enabled

Designer/Thread: WordLoggerDesigner

Priority: Normal
Startup Squelch (seconds): 0

☐ Has Timer
Timer Interval (milliseconds): 0

☒ Has WatchDog
WatchDog Timeout (seconds): 30

Processing Queue:

Length: 10000

When queue is full :

☒ Ignore new items
☐ Remove old items

Apply

Reset

Add New

Designer Properties

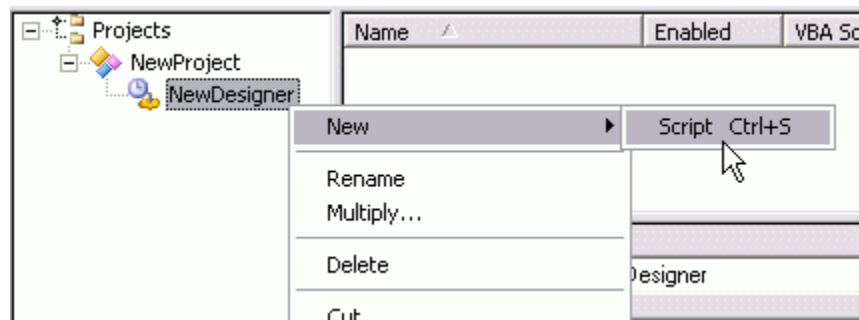
Scripts

Scripts link a particular subroutine within the designer to a trigger created with the Unified Data Manager. Whenever the trigger happens for a script, the associated subroutine will enter the queue for the designer/thread.

Creating a New Script

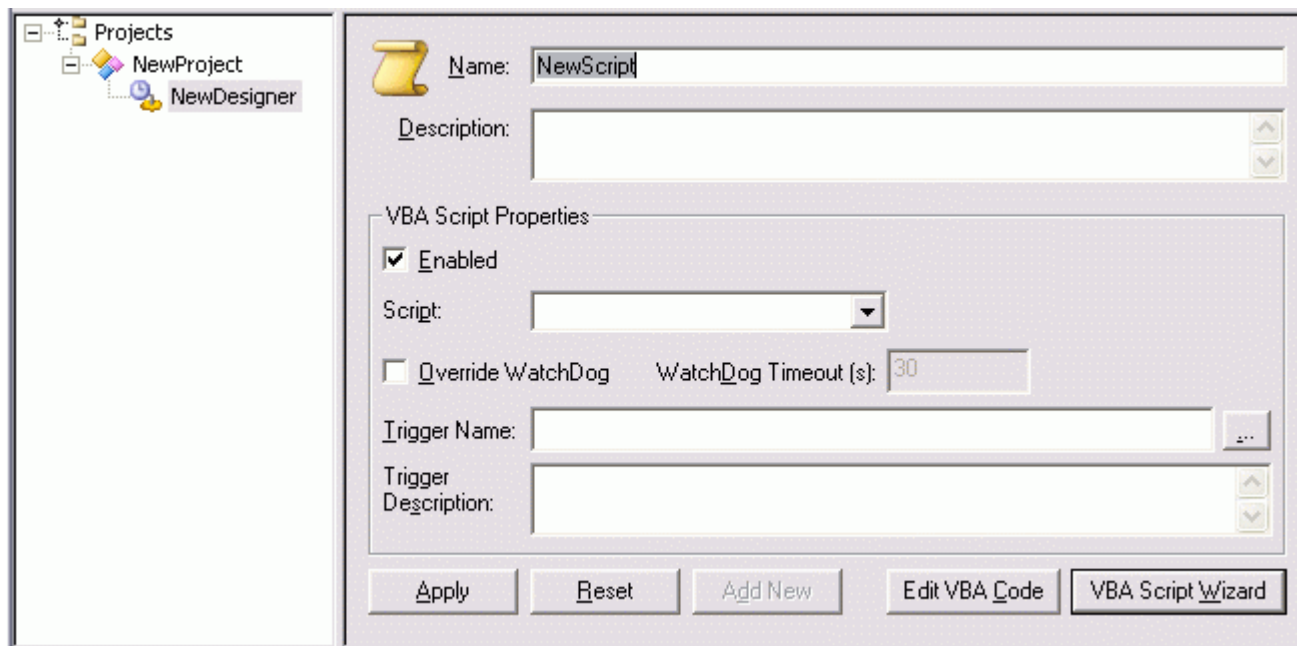
To create a new script:

1. Right-click on a designer/thread in the tree control of the Configurator and select **New > Script** from the pop-up menu, as shown in the figure below.



Creating a New Script

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



Setting the Properties for the New Script

3. In the **Name** field, type a name for the new script.
4. In the **Description** field, you can type an optional description for the new script.

- When you have finished configuring script properties, click the **Apply** button. The new script appears under the selected designer/thread in the tree control.

Script Properties

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

Name	Enabled	VBA Script	Trigger Name
LogToWord	Yes	LogToWord	trg_data:RandomTimeSine

Name: LogToWord

Description:

VBA Script Properties

☒ **Enabled**

Script: LogToWord

☐ **Override WatchDog** WatchDog Timeout (s): 30

Trigger Name: trg_data:RandomTimeSine

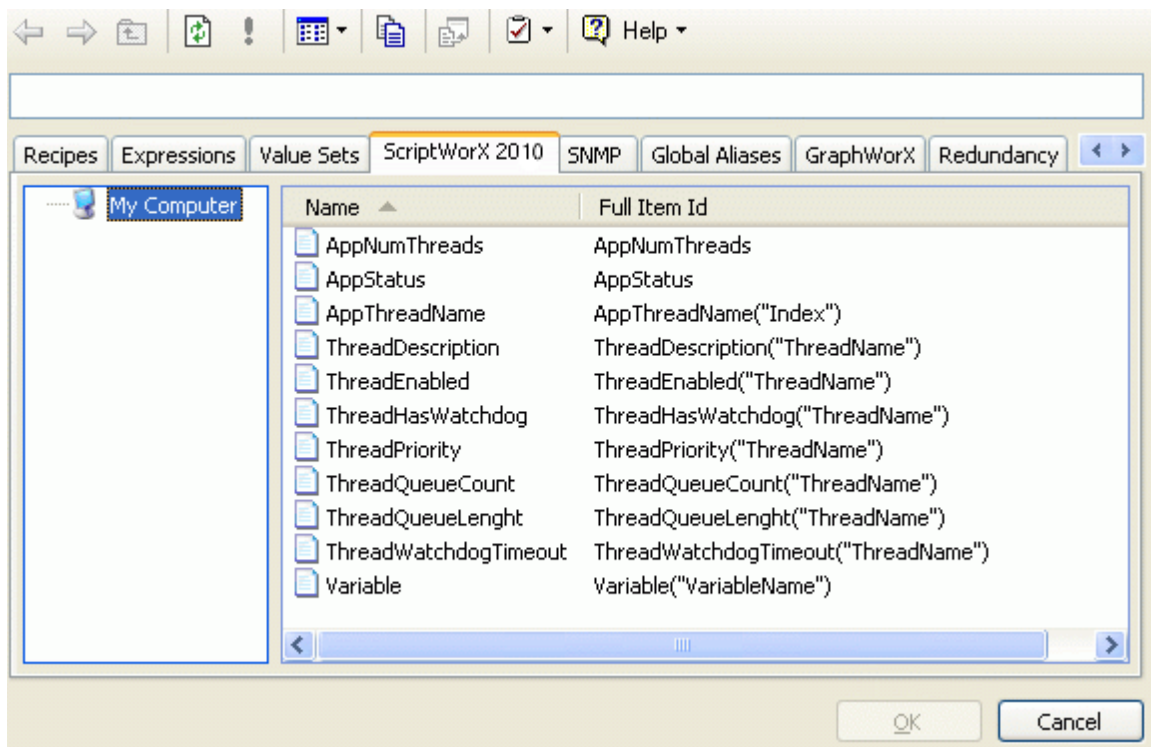
Trigger Description:

Buttons: Apply, Reset, Add New, Edit VBA Code, VBA Script Wizard

Script Properties

- Name:** Enter a name for the script.
- Description:** Enter a description for the script (optional).
- Enabled:** Check this check box to enable the script to run.
- Script:** Select a script type from the drop-down list. With the **OnTimer** event per thread, scripts execute per millisecond within the thread. With the **OnValueChange** event per thread, scripts execute when a global variable is modified.
- Override Watchdog:** Ignores the default watchdog timeout period defined in the designer/thread properties.
- Watchdog Timeout:** When **Override Watchdog** is checked, you can specify a watchdog timeout that applies only to this script, which will ignore the default watchdog timeout period defined in the designer/thread properties.

- **Trigger Name:** Clicking the ... button opens the Unified Data Browser, which allows you to browse for global triggers (configured in the Unified Data Manager). Select a trigger to use for your script, as shown in the figure below.
- **Trigger Description:** Enter a description for the trigger (optional).



Selecting Triggers From the Unified Data Browser

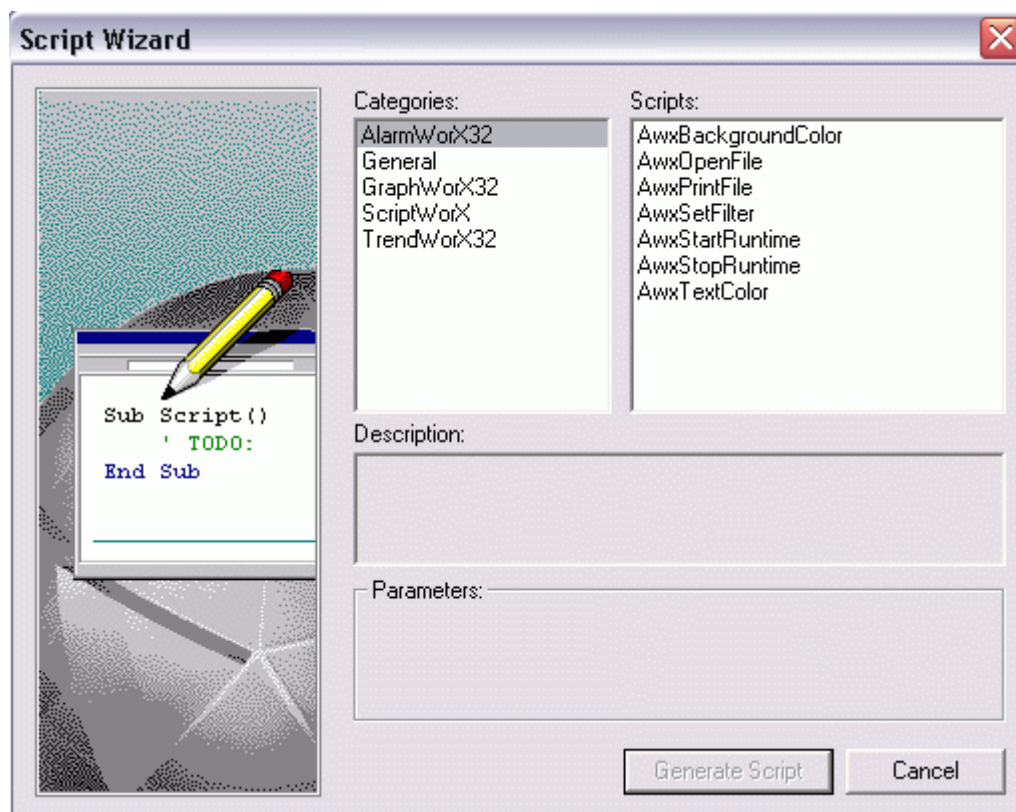
- **Edit VBA Code:** Launches the integrated Microsoft Visual Basic Editor, which allows you to write, edit, and debug your scripts.
- **VBA Script Wizard:** Launches the Visual Basic for Applications Script Wizard, which contains many script templates for the most common operations. For more information, please see the "VBA Script Wizard" section below.
- **Apply:** Saves all changes to the configuration database.
- **Reset:** Clears all recently added information.
- **Add New:** Creates a new designer to be configured.

VBA Script Wizard

The **VBA Script Wizard** is a tool that is available from each script configuration window and that allows you to generate scripts from script templates. Each script template is stored in one script template file (.stp), which is located in the Script Wizard Template directory placed in the installation directory of ScriptWorX32. Script templates support several keywords, which are replaced during script generation by values entered by in the Wizard edit fields. Script templates also support the addition of necessary TypeLib references that can be appended to the VBA container. TypeLib references and optional keywords must be specified in the header section of the script.

Running the Script Wizard

To run the ScriptWorX VBA Script Wizard, shown in the figure below, click the **VBA Script Wizard** button on the script properties configuration screen.



Script Wizard

The Script Wizard provides several categories of script templates from which to generate a script.

Description

The **Description** section tells you what the script is supposed to be doing and what components need to be installed on your system before the script will run properly. The text that is shown in this section is the text that was entered in the VBA code after the **#Description** keyword.

Parameters

Depending on the category and the script selected, several items are available in the **Parameters** section of the window. The parameters available are set in the configuration of the script template by using the **#Parameters** keyword. The entries for each of these fields will be directly inserted into the script and will determine exactly how the scripts run.

Installed Script Categories

There are five installed script template categories: AlarmWorX32, General, GraphWorX32, ScriptWorX, and TrendWorX32. Each of these categories includes associated scripts. Each script includes a description of what the script does and what components must be installed.

AlarmWorX32 Script Categories

Script	Description
AwxBackgroundColor	Set background color of AlarmWorX32 Viewer ActiveX. GraphWorX32 and AlarmWorX32 Viewer ActiveX must be installed.
AwxOpenFile	Open GraphWorX32 display with an AlarmWorX32 Viewer ActiveX and load the AlarmWorX32 Viewer ActiveX configuration file.
AwxPrintFile	Print the display containing the AlarmWorX32 Viewer ActiveX. GraphWorX32 and AlarmWorX32 Viewer ActiveX must be installed.
AwxSetFilter	Set filter name and filter expression for AlarmWorX32 Viewer ActiveX. GraphWorX32 and AlarmWorX32 Viewer ActiveX must be installed.
AwxStartRuntime	Enter Runtime Mode. GraphWorX32 and AlarmWorX32 Viewer ActiveX must be installed.
AwxStopRuntime	Exit Runtime Mode. GraphWorX32 and AlarmWorX32 Viewer ActiveX must be installed.
AwxTextColor	Set text color of AlarmWorX32 Viewer ActiveX. GraphWorX32 and AlarmWorX32 Viewer ActiveX must be installed.

General Script Categories

Script	Description
Beeper	The script Beeps when started.
DiskSpace	The script obtains total and free disk space.
HelloICONICS	The script pops up a message box that says "Hello from ICONICS."
Keypad	The script displays a Keypad form for entering a text string.
Memory	The script obtains total and available physical memory.
Numpad	The script displays a Numpad form for entering a double value.

GraphWorX32 Script Categories

Script	Description
GwxArrayOfEllipses	Script starts GraphWorX32 and creates an array of ellipses, where the array dimensions and ellipse properties are specified.
GwxArrayOfRectangles	Script starts GraphWorX32 and creates an array of rectangles, where the array dimensions and rectangle properties are specified.
GwxClosePopupWindow	Script starts GraphWorX32 and opens the popup window specified in the File Name field. Then the popup window is closed.

GwxEmbeddedWindow	Script starts GraphWorX32 and opens the display selected in the File Name field as an embedded window.
GwxIterateObjects	Iterates all symbols and their subsymbols in the display.
GwxMaximizeWindow	Script starts GraphWorX32 and maximizes its window. GraphWorX32 must be installed.
GwxMinimizeWindow	Script starts GraphWorX32 and minimizes its window. GraphWorX32 must be installed.
GwxOpenDisplay	Script starts GraphWorX32 and opens the display selected in the File Name field.
GwxPopupWindow	Script starts GraphWorX32 and opens a popup window specified in the File Name field. Use the Browse button.
GwxPrintDisplay	Script starts GraphWorX32, and opens and prints the display selected in the File Name field.
GwxSetBackgroundColor	Script starts GraphWorX32 and changes the display background color.
GwxSetDisplayDimensions	Script starts GraphWorX32 and sets the display dimensions (work area/world bounds) of the currently loaded display.
GwxSetWindowDimensions	Script starts GraphWorX32 and sets the GraphWorX32 main window size and location as a percentage of the total screen size.
GwxStartRuntime	Script starts GraphWorX32 and switches to runtime mode. GraphWorX32 must be installed.
GwxStopRuntime	Script starts GraphWorX32 and stops runtime mode.
GwxToggleRuntime	Script toggles GraphWorX32 runtime mode. GraphWorX32 must be installed.

ScriptWorX Script Categories

Script	Description
InfiniteScript	Implements body of an "infinite" script. It loops and waits for the ScriptWorX shut-down event.
QueueSource	Queue source implementation with error handling.
QueueTarget	Queue target implementation with error handling – an infinite script with queue data reading.
ReadOPCTag	Read an OPC tag using IDataPoint object; waits for the first update.
ReadOPCTag_NoWait	The same as "ReadOPCTag" script, but it does not wait for the first update.
WriteOPCTag	Write an OPC tag using IDataPoint object.

TrendWorX32 Script Categories

Script	Description
TwxAddTrend	Adds real-time trend to the TrendWorX32 Viewer. GraphWorX32 and TrendWorX32 Viewer must be installed.
TwxBackground Color	Sets TrendWorX32 Viewer background color. GraphWorX32 and TrendWorX32 Viewer must be installed.
TwxDeleteTrend	Deletes real-time trend pen. GraphWorX32 and TrendWorX32 Viewer must be installed.
TwxOpenFile	Script starts GraphWorX32 display with a TrendWorX32 Viewer ActiveX. GraphWorX32 and TrendWorX32 Viewer must be installed.
TwxPrintFile	Prints the display containing the TrendWorX32 Viewer ActiveX. GraphWorX32 and TrendWorX32 Viewer must be installed.
TwxShowDetails	Shows or hides the detailed list of trends. GraphWorX32 and TrendWorX32 Viewer must be installed.
TwxShowTitle	Shows or hides the TrendWorX32 Viewer title. GraphWorX32 and TrendWorX32 Viewer must be installed.
TwxStartRuntime	Enters runtime mode. GraphWorX32 and TrendWorX32 Viewer must be installed.
TwxStopRuntime	Exits runtime mode. GraphWorX32 and TrendWorX32 Viewer must be installed.

Using a Template

To access a script template, the **VBA Script Wizard** button in the script properties configuration. This will open the Script Wizard as described above.

Select the desired category and select the appropriate script from the list.

Click on the script. The description of the script should appear in the **Description** field.

Once you have set up the Wizard as desired, click the **Generate Script** button. The ScriptWorX user interface will be restored, and you will be asked if you want to view the VBA code.

ScriptWorX 2010 Examples

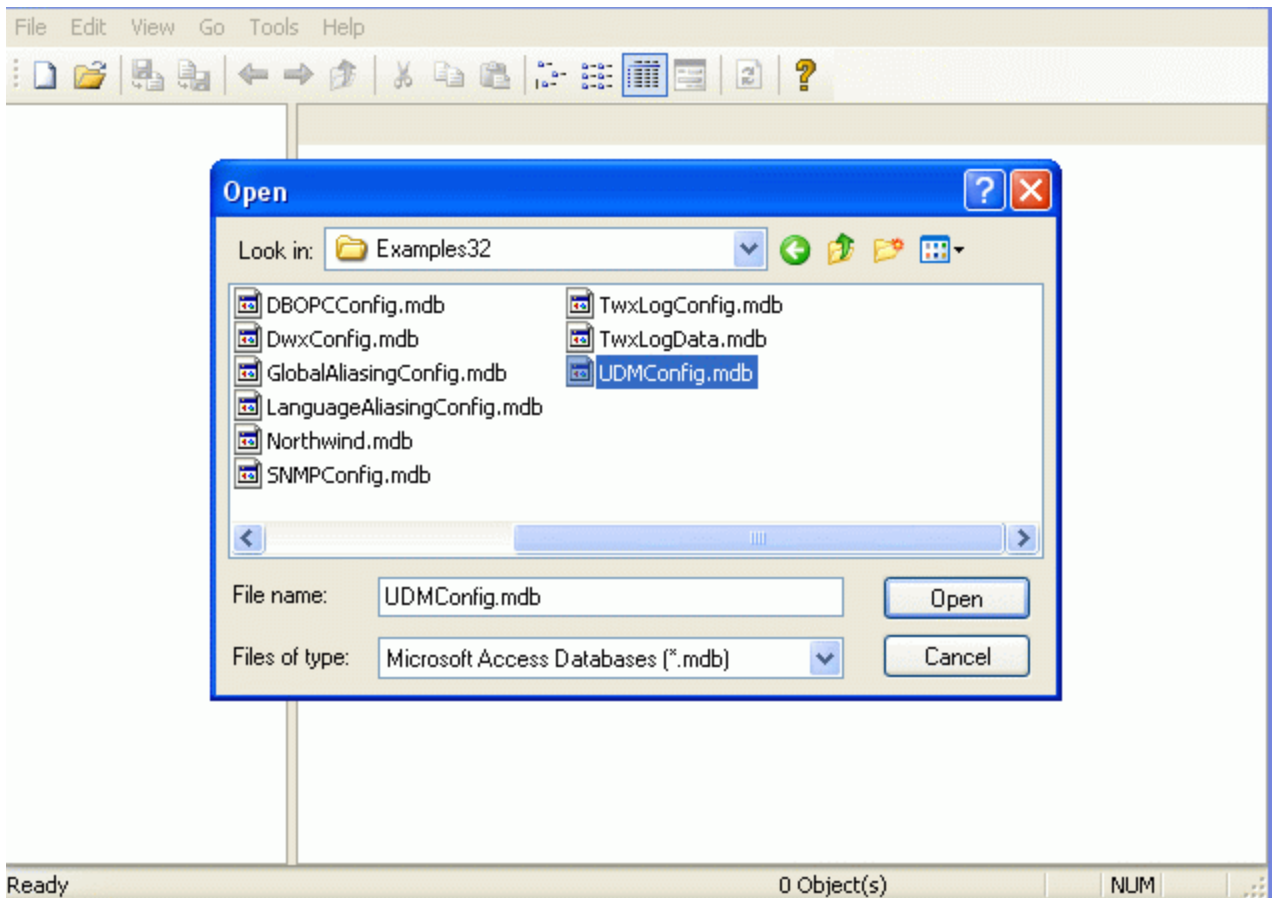
This section describes a few of the ScriptWorX 2010 example projects that are installed with GENESIS32. To you these examples, you must have a working knowledge of GENESIS32 and VBA.

The ScriptWorX 2010 examples are located in the "Program Files\ICONICS\GENESIS32\Examples\ScriptWorX2010" installation folder.

Activating the Unified Data Manager Database

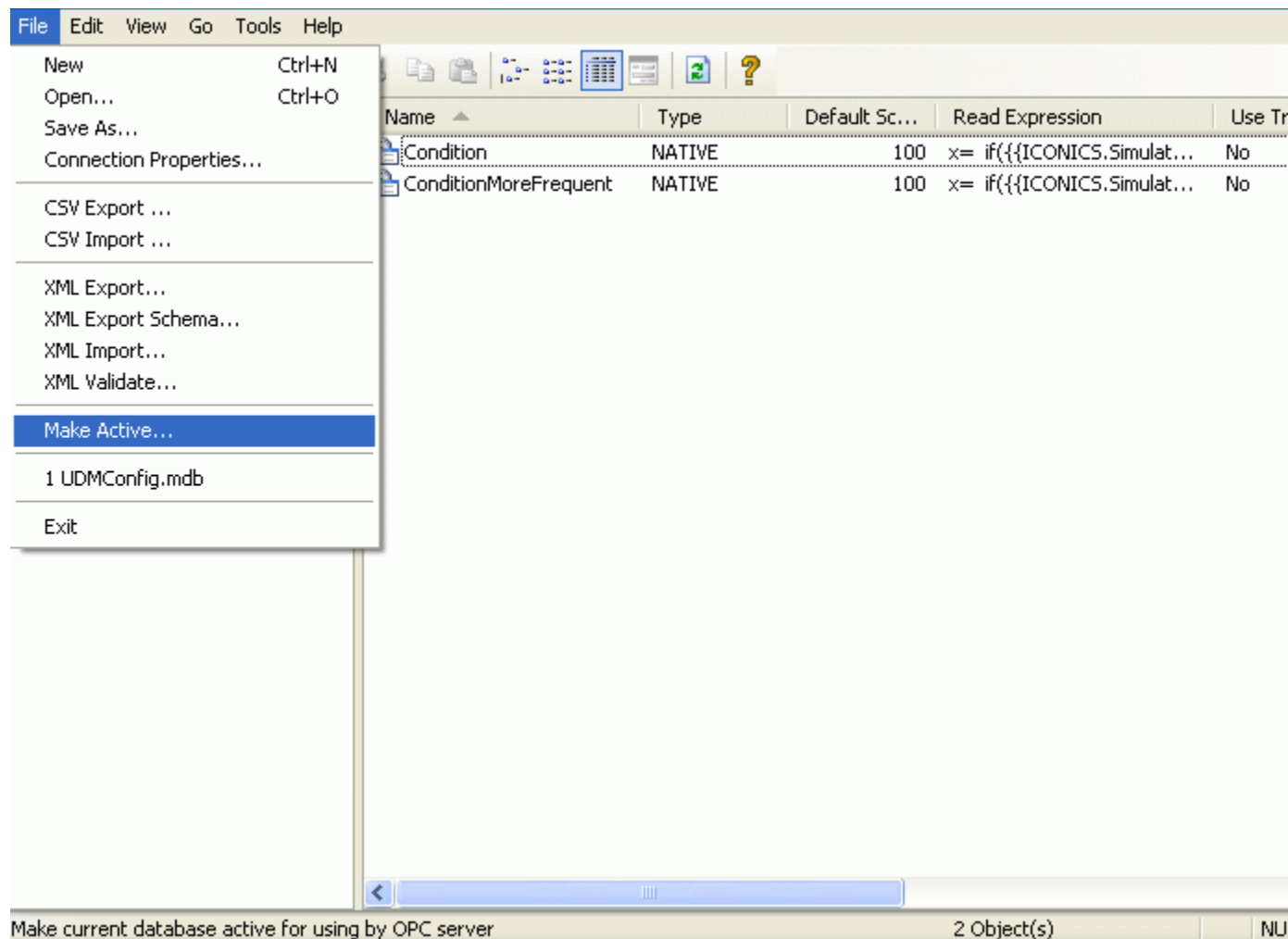
To run the ScriptWorX 2010 examples, you must first activate the Unified Data Manager configuration database:

1. To launch the Unified Data Manager from the Windows **Start** menu, select **Programs > ICONICS GENESIS32 Unified Data Manager**.
2. Select **Open** from the **File** menu, and then browse to the "\\Documents and Settings\\All Users\\ICONICS\\Examples32" directory. Open the **UDMConfig.mdb** database, as shown in the figure below.



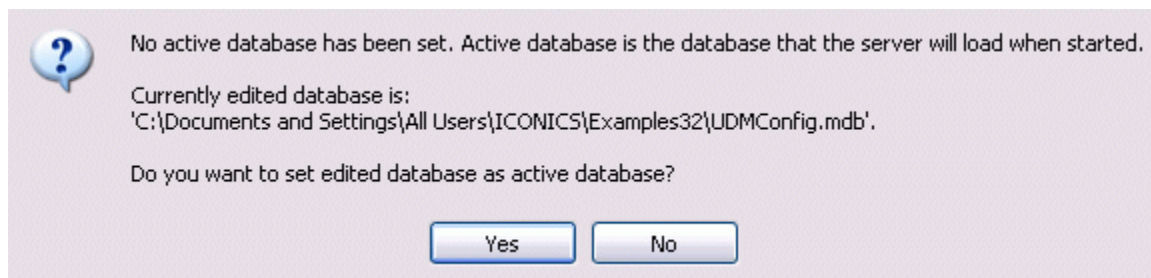
Opening the Unified Data Manager Database

3. To activate the database, select **Make Active** from the **File** menu, as shown in the figure below.



Activating the Database in the Unified Data Manager

4. Click **Yes** to activate the database, as shown in the figure below.



Activating the Database

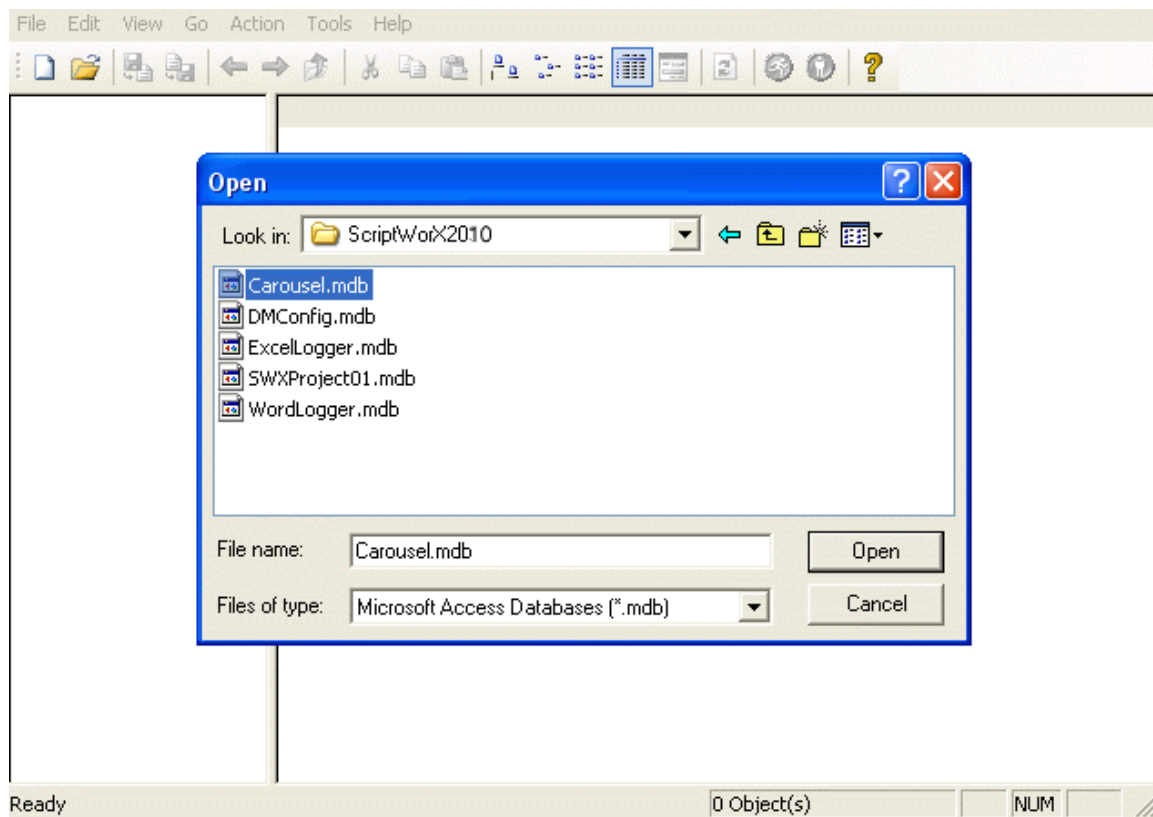
Carousel Example

The **Carousel** example opens a series of GraphWorX32 display files every time the script is run. The displays to be shown are listed in the file "Carousel.txt" file, which is located in the "\Program Files\ICONICS\GENESIS32\Examples\GEN32DEMO" directory. Notice that the files listed do not have a path. The example reads a line from the file and pastes the path of GEN32DEMO in front of the line. This means that all displays that you want to add to the Carousel must be located in the "\Program Files\ICONICS\GENESIS32\Examples\GEN32DEMO" directory.

To use this example, you must have GraphWorX32 and the GEN32DEMO examples installed on your computer. Make sure that the GraphWorX32 Object is selected in the **Tools > References** menu in the VBA Editor.

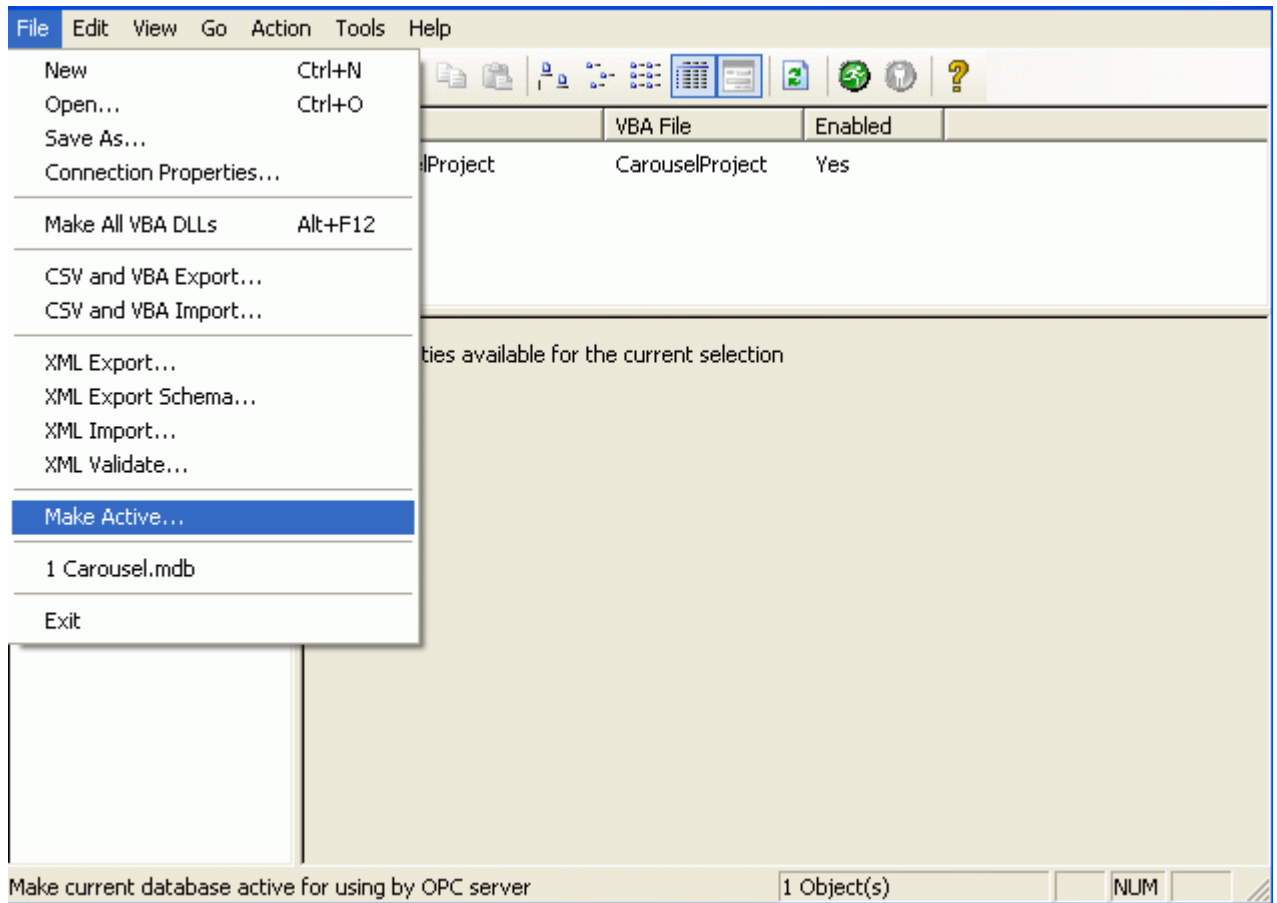
To run the Carousel example:

1. Launch the ScriptWorX 2010 Configurator from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > ScriptWorX > ScriptWorX 2010**.
2. Select **Open** from the **File** menu, and then browse to the "\Program Files\ICONICS\GENESIS32\Examples\ScriptWorX2010" directory. Open the **Carousel.mdb** database, as shown in the figure below.



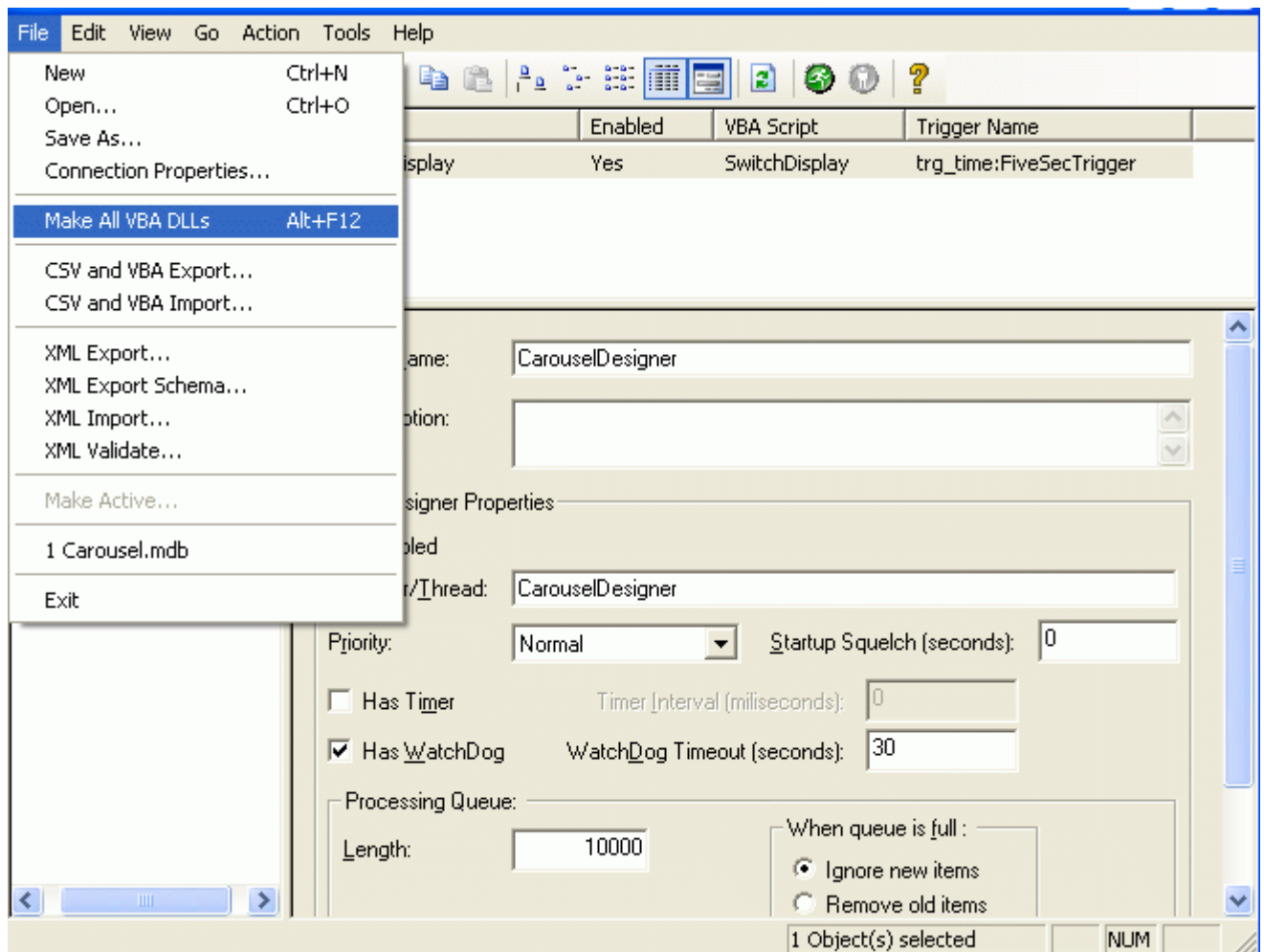
Opening the Carousel Example in ScriptWorX 2010

3. To activate the database, select **Make Active** from the **File** menu, as shown in the figure below. Click **Yes** to activate the database.



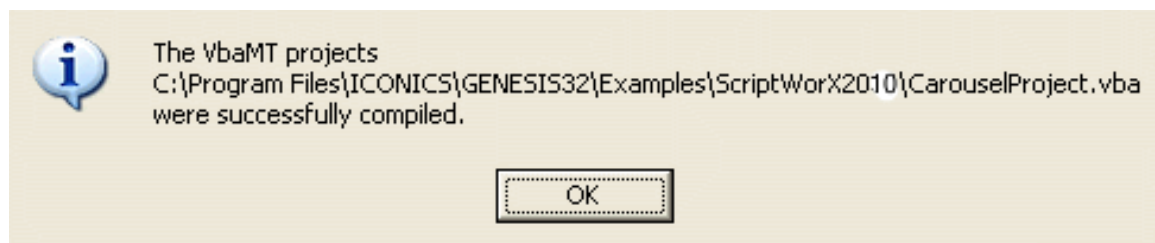
Activating the Carousel.mdb Database

4. Select **Make All VBA DLLs** from the **File** menu, as shown in the figure below.



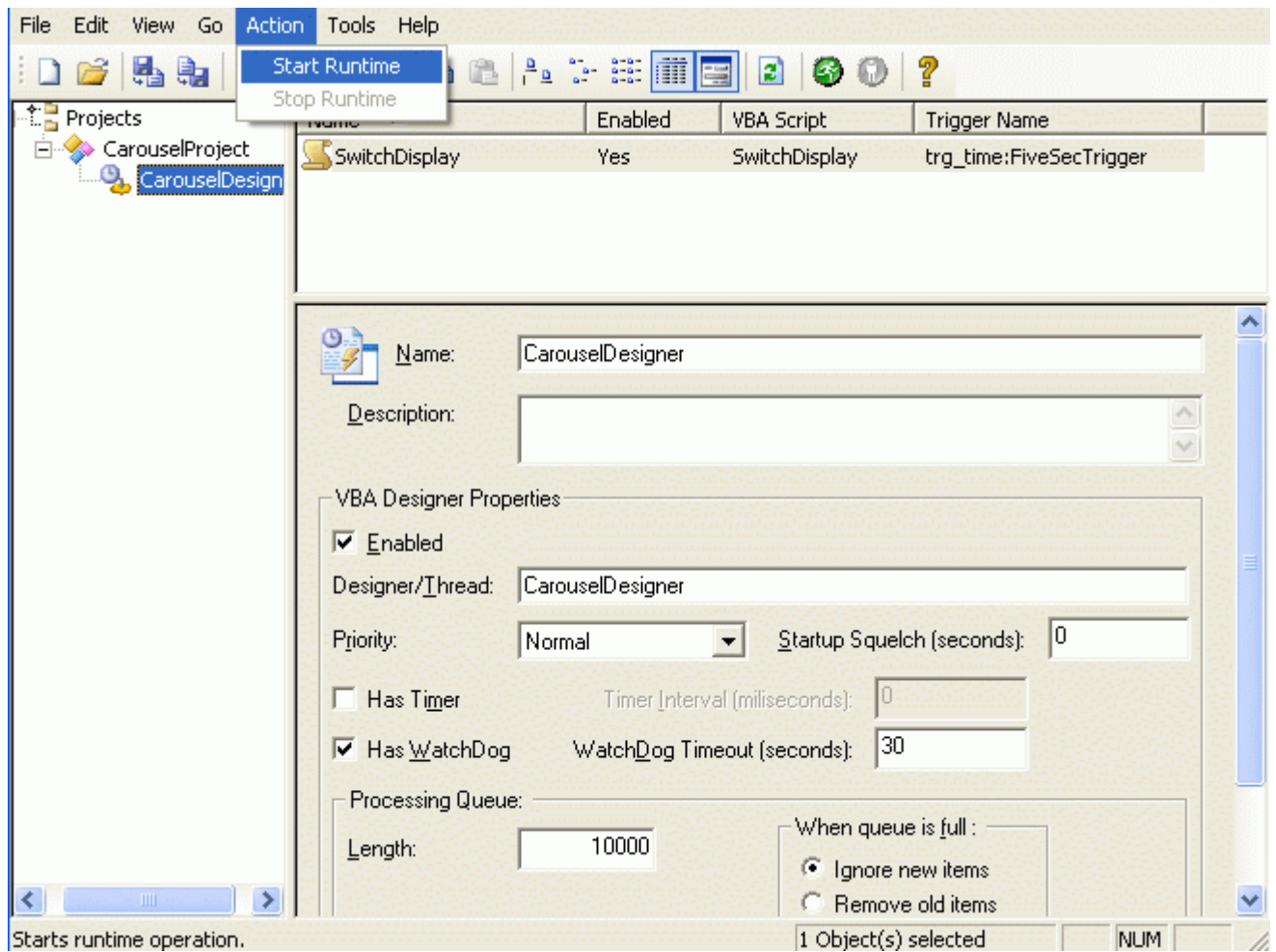
Compiling the Carousel Project

5. The **CarouselProject.vba** project will be compiled, and a confirmation message will appear, as shown in the figure below.



Project Successfully Compiled

6. Enter the project into runtime mode by selecting **Start Runtime** from the **Action** menu, as shown in the figure below. The Carousel example script launches the series of GraphWorX32 display files.



Starting Runtime Mode

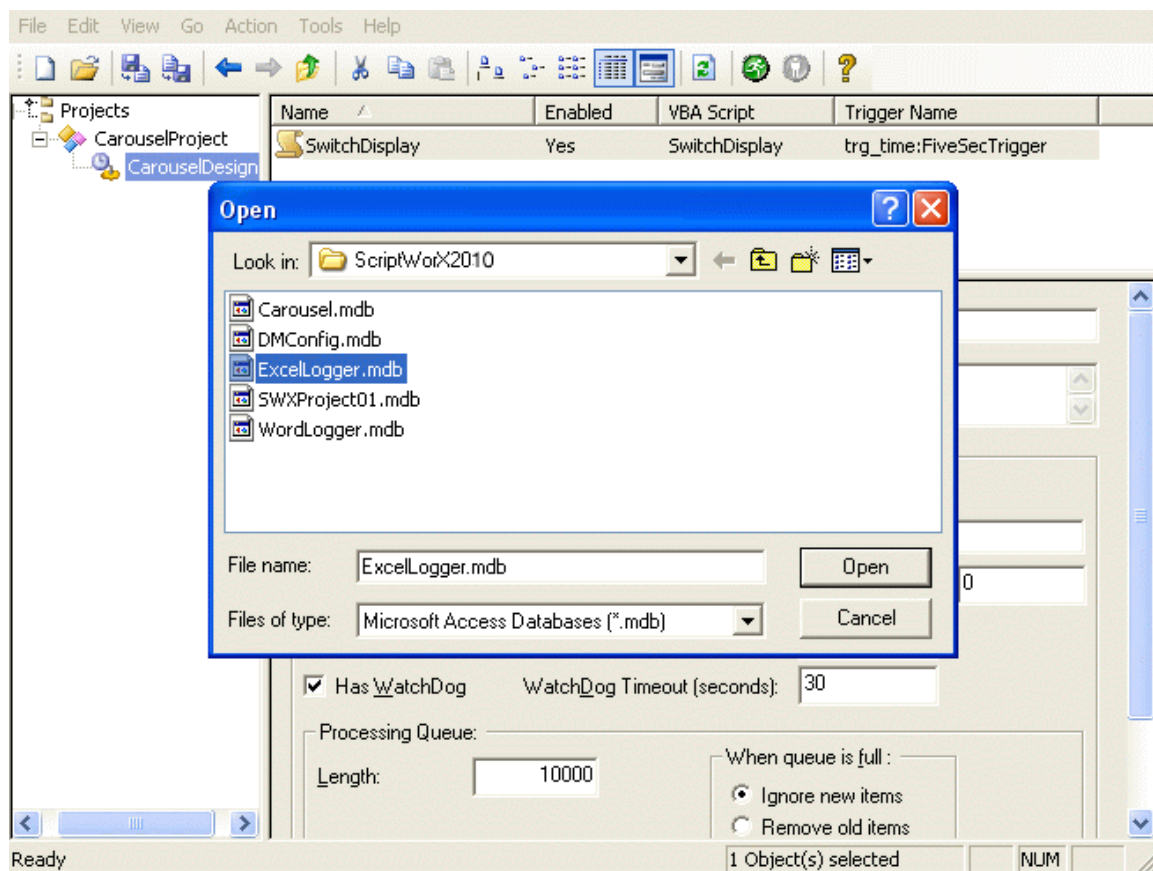
Excel Logger Example

The **Excel Logger** example writes OPC tag values to Microsoft Excel using the Excel OLE Automation Interface. The example is currently set up to log the OPC Simulator "Sine" tag, which it receives as an argument from "TriggerData".

To use this example, you must have Microsoft Excel 97 or higher installed on your computer. Make sure that the Microsoft Excel Object (version 8.0 or higher) is selected in the **Tools > References** menu in the VBA Editor.

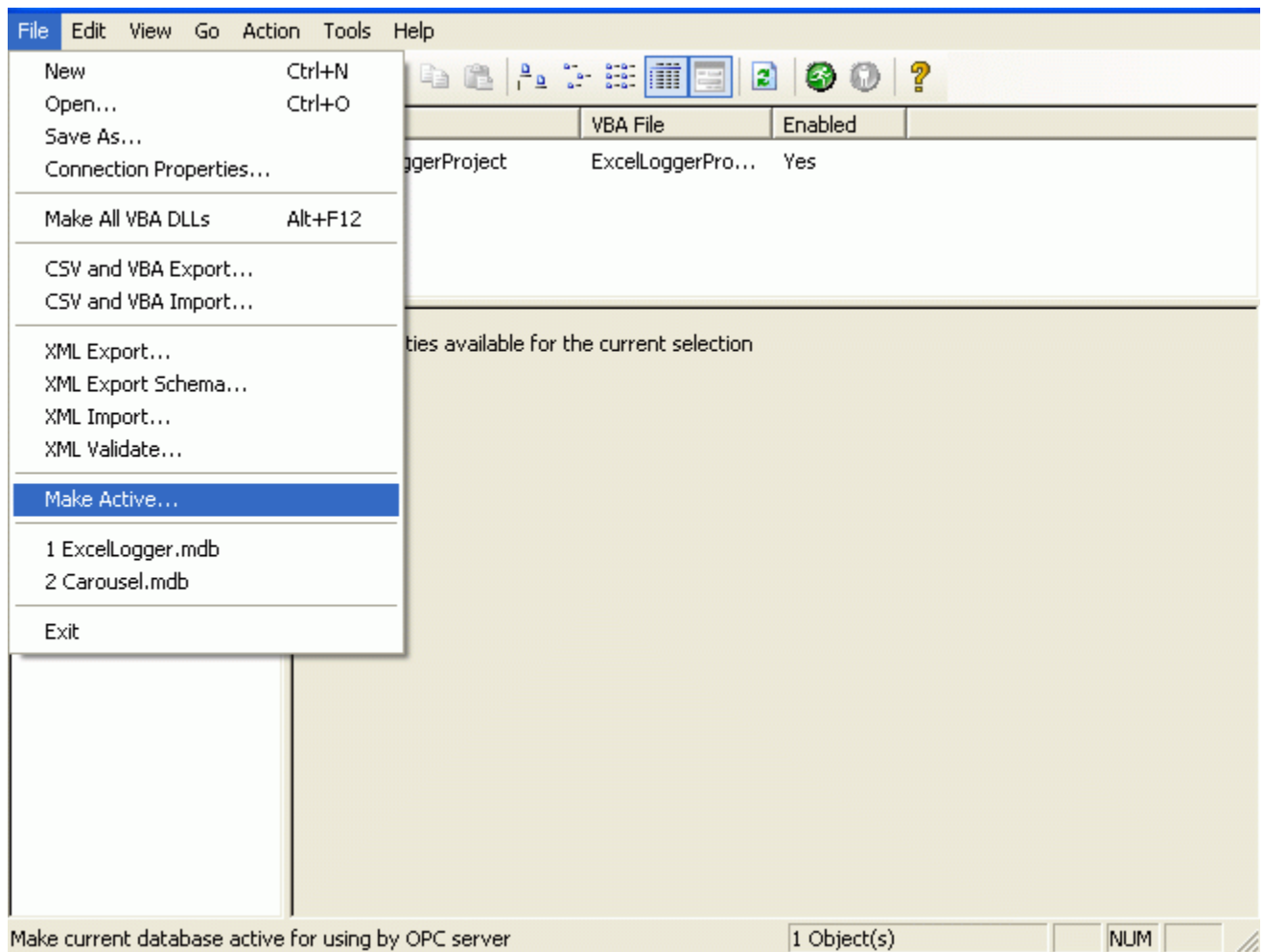
To run the Excel Logger example:

1. Launch the ScriptWorX 2010 Configurator from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > ScriptWorX > ScriptWorX 2010**.
2. Select **Open** from the **File** menu, and then browse to the "\\Program Files\\ICONICS\\GENESIS32\\Examples\\ScriptWorX2010" directory. Open the **ExcelLogger.mdb** database, as shown in the figure below.



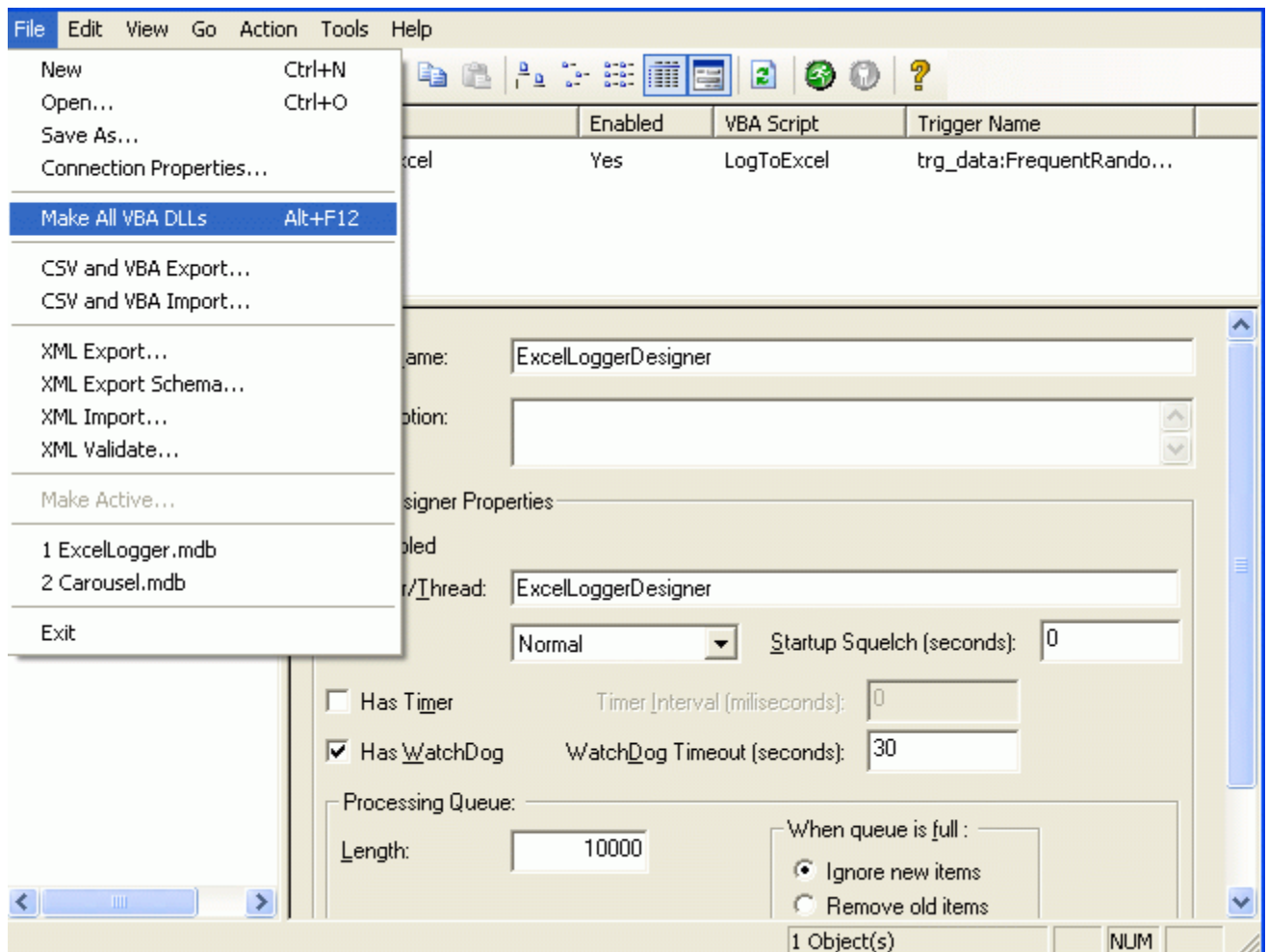
Opening the Excel Logger Example in ScriptWorX 2010

3. To activate the database, select **Make Active** from the **File** menu, as shown in the figure below. Click **Yes** to activate the database.



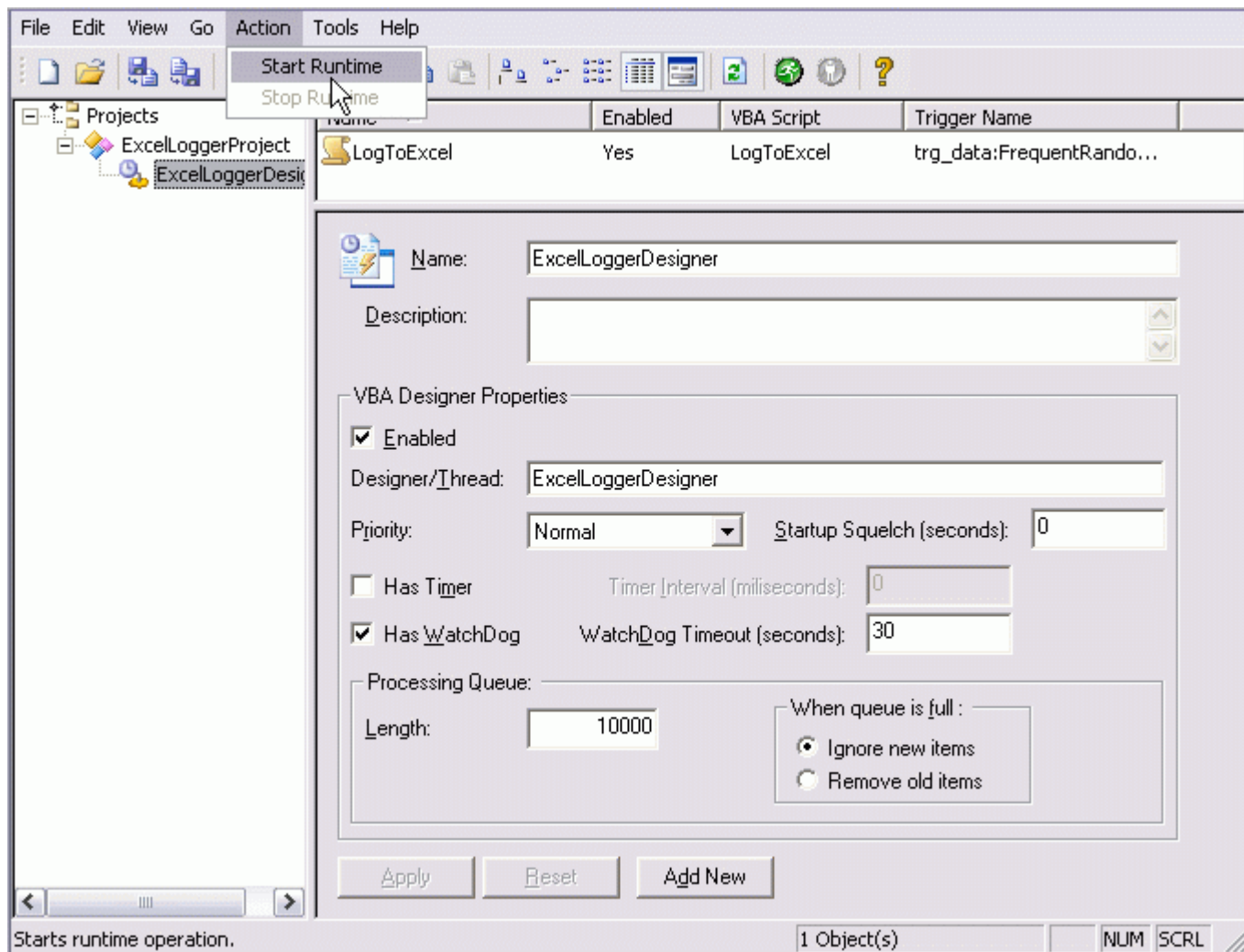
Activating the ExcelLogger.mdb Database

4. Select **Make All VBA DLLs** from the **File** menu, as shown in the figure below. The **ExcelLoggerProject.vba** project will be compiled.



Compiling the Excel Logger Project

5. Enter the project into runtime mode by selecting **Start Runtime** from the **Action** menu, as shown in the figure below.



Starting Runtime Mode

- The script launches an Excel spreadsheet that displays the tag values and quality status, as shown in the figure below.

Microsoft Excel - Sheet1

File Edit View Insert Format Tools Data Window Help

Type a question for help

C3 Quality

Center

ScriptWorX32 Logging Results

	A	B	C	D	E	F	G
1							
2							
3	Timestamp	Tag Value	Quality				
4	4/27/2006 22:39	-0.269860834	192				
5	4/27/2006 22:39	-0.3357144	192				
6	4/27/2006 22:39	-0.578540921	192				
7	4/27/2006 22:39	-0.778418303	192				
8	4/27/2006 22:39	-0.819552124	192				
9	4/27/2006 22:39	-0.944422185	192				
10	4/27/2006 22:39	-0.938758016	192				
11	4/27/2006 22:39	0.353996992	192				
12	4/27/2006 22:40	0.994022071	192				
13	4/27/2006 22:40	-0.87657547	192				
14	4/27/2006 22:40	-0.934303939	192				
15							
16							
17							
18							
19							
20							
21							
22							

Logger Data

Ready

NUM

Running the Excel Logger Project in Microsoft Excel

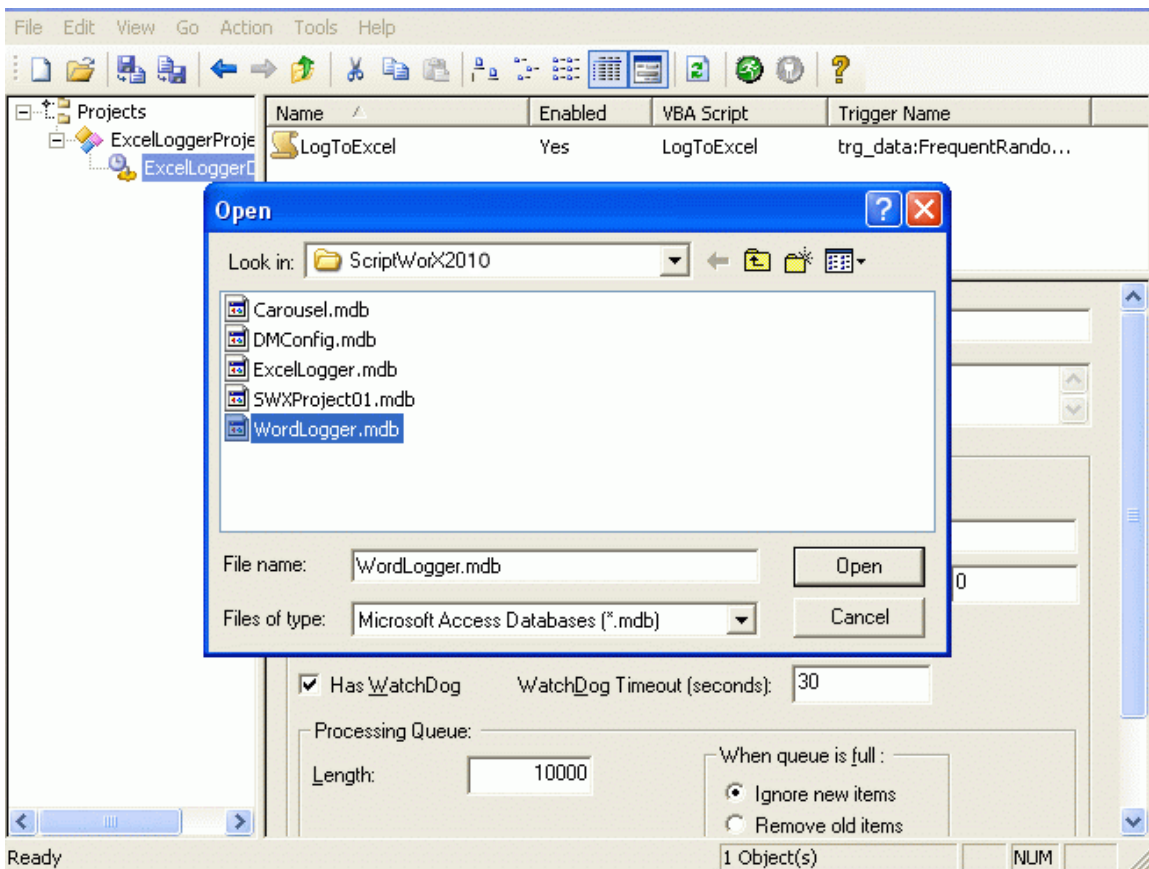
Word Logger Example

The **Word Logger** example writes OPC tag values to Microsoft Word using the Word OLE Automation Interface. The example is currently set up to log the OPC Simulator "Sine" tag obtained as "TriggerData" from the Unified Data Manager.

To use this example, you must have Microsoft Word 97 or higher installed on your computer. Make sure that the Microsoft Word Object (version 8.0 or higher) is selected in the **Tools > References** menu in the VBA Editor.

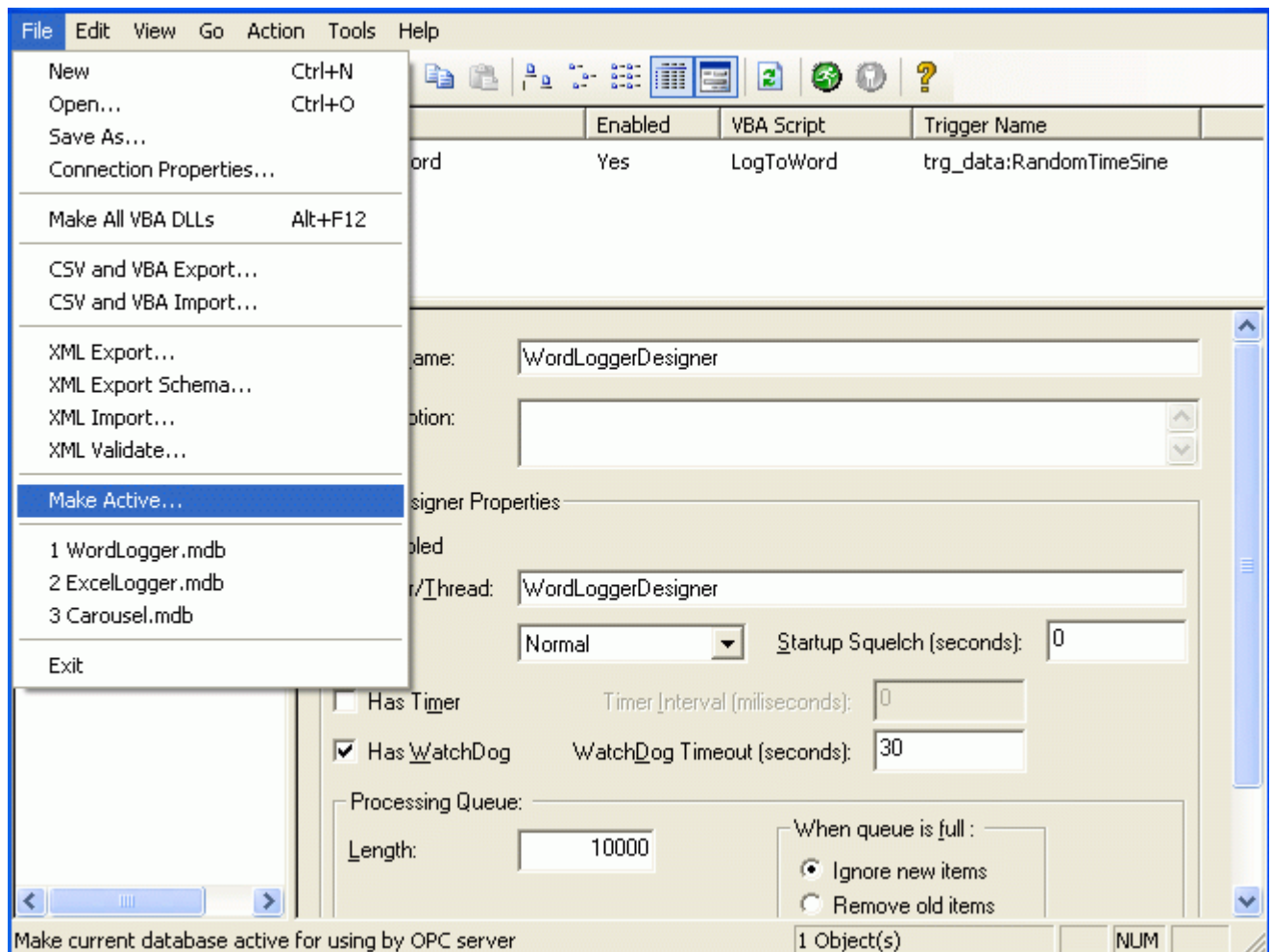
To run the Word Logger example:

1. Launch the ScriptWorX 2010 Configurator from the Windows **Start** menu by selecting **Programs > ICONICS GENESIS32 > ScriptWorX > ScriptWorX 2010**.
2. Select **Open** from the **File** menu, and then browse to the "\\Program Files\\ICONICS\\GENESIS32\\Examples\\ScriptWorX2010" directory. Open the **WordLogger.mdb** database, as shown in the figure below.



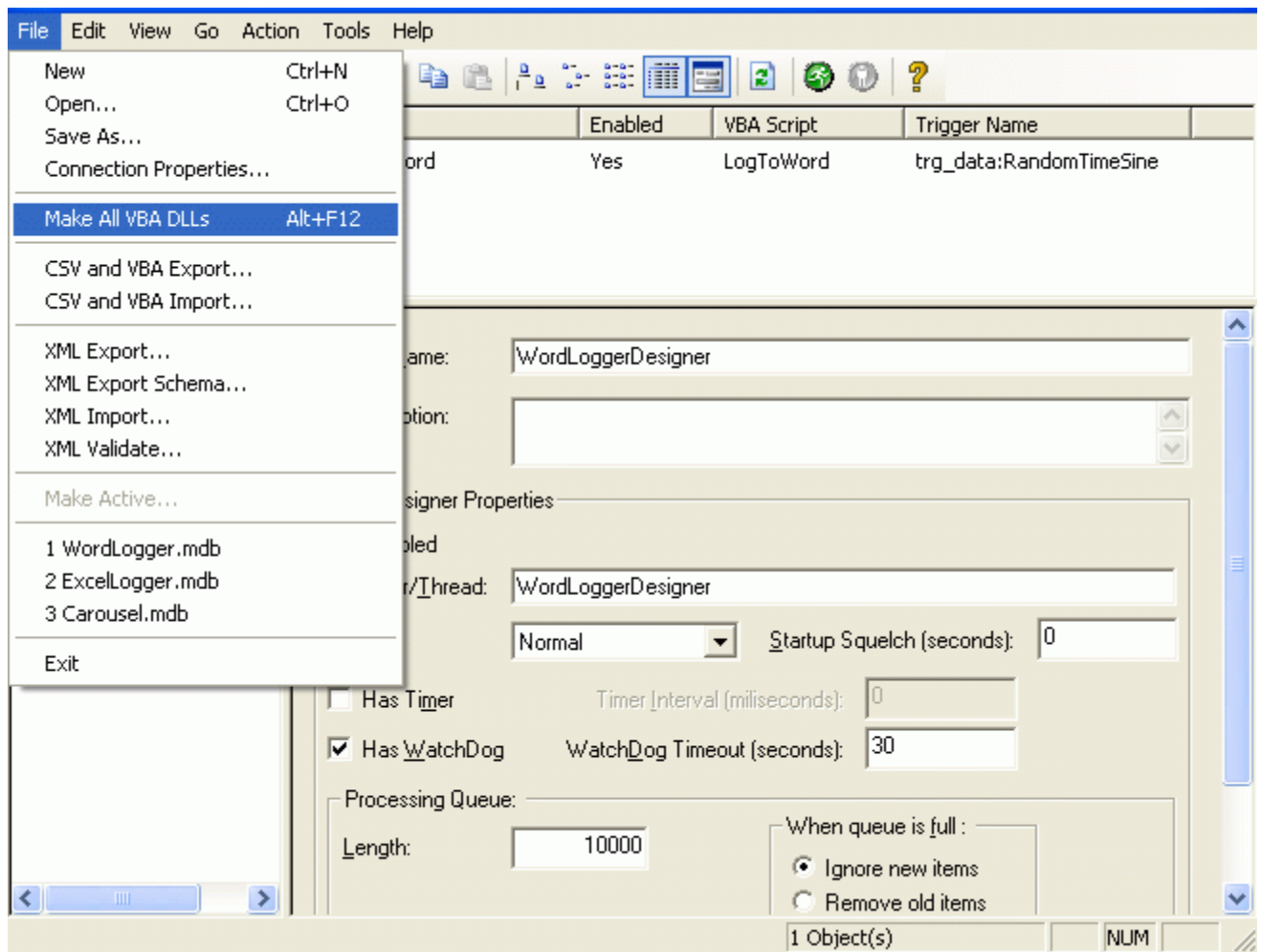
Opening the Word Logger Example in ScriptWorX 2010

3. To activate the database, select **Make Active** from the **File** menu, as shown in the figure below. Click **Yes** to activate the database.



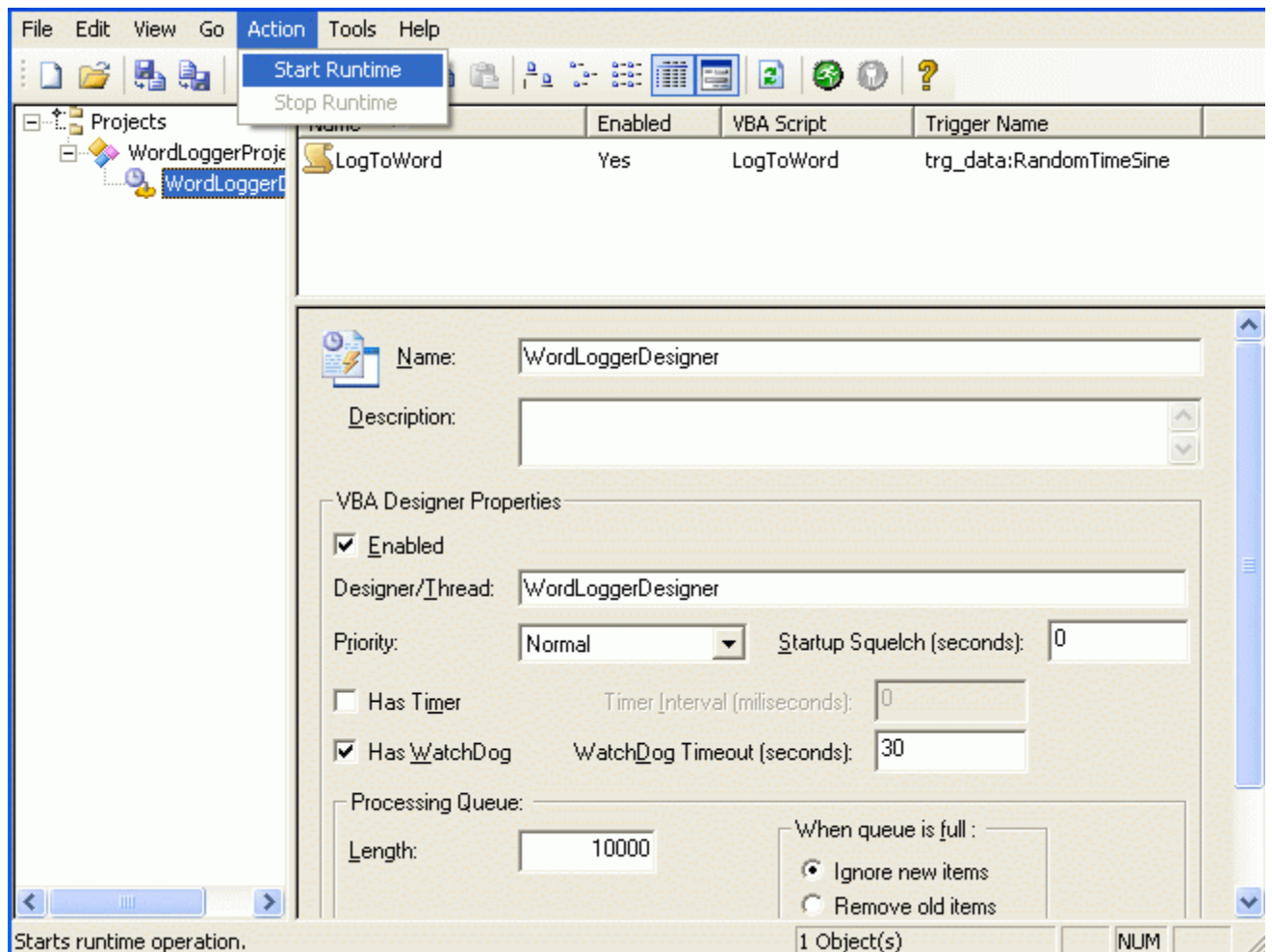
Activating the WordLogger.mdb Database

4. Select **Make All VBA DLLs** from the **File** menu, as shown in the figure below. The **WordLoggerProject.vba** project will be compiled.



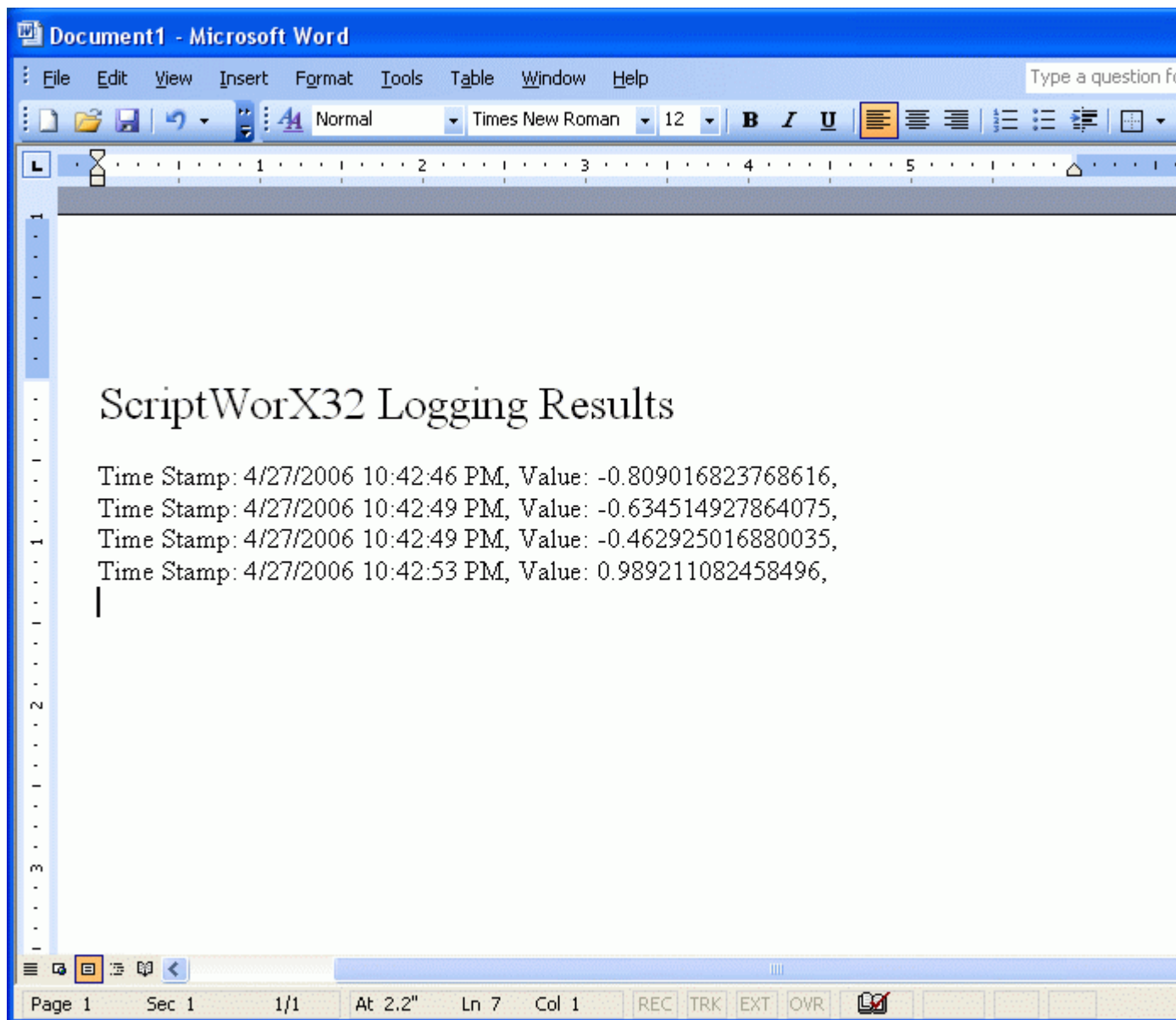
Compiling the Word Logger Project

5. Enter the project into runtime mode by selecting **Start Runtime** from the **Action** menu, as shown in the figure below.



Starting Runtime Mode

- The script launches a Word document that displays the tag values and quality status, as shown in the figure below.



Running the Word Logger Project in Microsoft Word