

# Debugging Validation (Index) Scripts With Visual Studio .NET

## Application Note

<b>Date</b>	August 24, 2011
<b>Applies To</b>	Kofax Capture 8.0, 9.0, 10.0
<b>Summary</b>	This application note provides the information needed to step through a .NET-based Validation Script during execution.
<b>Revision</b>	1.1

### Overview

When developing and debugging Validation Scripts (also referred to as Index Scripts, as they are also run by the Verification module), it is often desirable to step through the code while it is running.

This provides the ability to examine the actions taking place, along with values being used, in the code during execution. This information can be invaluable during development or troubleshooting any issues that may arrive at a later time.

This application note contains information on the requirements, preparation and basic steps a developer can use to step through the source code for a Validation Script using Visual Studio .NET. This Application Note specifically deals with the debugging of Validation Scripts created with Visual Studio .NET 2008 or later and does not address troubleshooting SBL-based scripts.

### Requirements

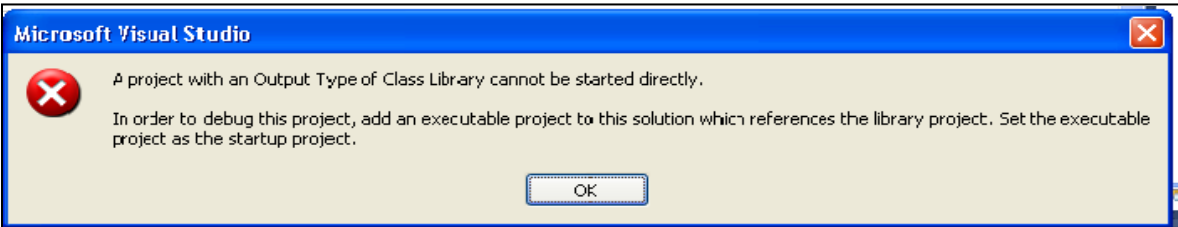
To implement the following procedures, you must have a Visual Studio .NET development environment installed on a machine with a Kofax Capture Standalone installation, a Capture Client installation or a Capture Server installation with client applications installed. Also, a batch containing documents from a Document Class making use of the Validation Script and ready to be processed by the Validation module (queue) will be needed.

### Stepping Through the Code

The most flexible method of examining what is happening in a Validation Script is to step through the code during execution.

### Setting the Validation Script to Automatically Launch Into Debug Mode

Visual Basic .NET Validation Scripts are compiled as Class Library DLLs and are dependent upon the COM object exposed and populated by the Validation module. As a consequence, they cannot be executed directly. If attempted, it will produce an error condition.



In order to step through the code of the script, it is necessary to attach to the Validation module (Index.exe) process through the Visual Studio IDE. The instructions to attach to the selected module, "existing process", are presented later in this Application Note in the section "Attaching to a Existing Process".

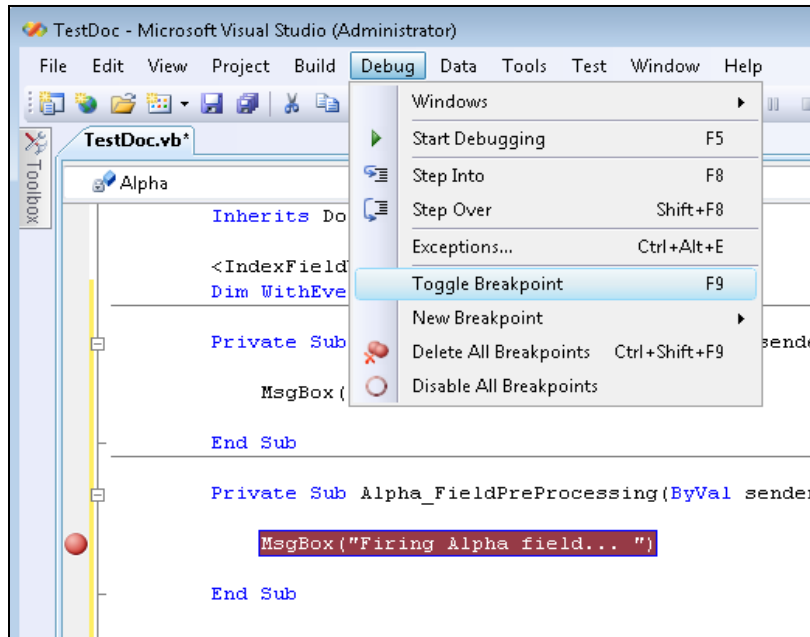
# Debugging Validation (Index) Scripts With Visual Studio .NET

*Application Note*

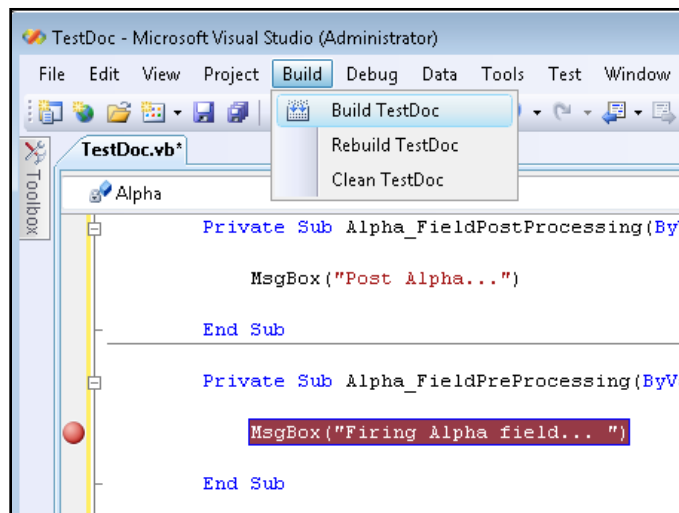
## Setting the Breakpoint

In the screenshot below, a breakpoint is set on a message box statement in the Pre Process function of an Index Field named "Alpha".

The breakpoint can be activated by clicking the left hand margin of the code window on the line you wish to set it or selecting "Toggle Breakpoint" from the Debug menu or pressing **F9**.



After the breakpoint has been created, build the project from the Build menu.

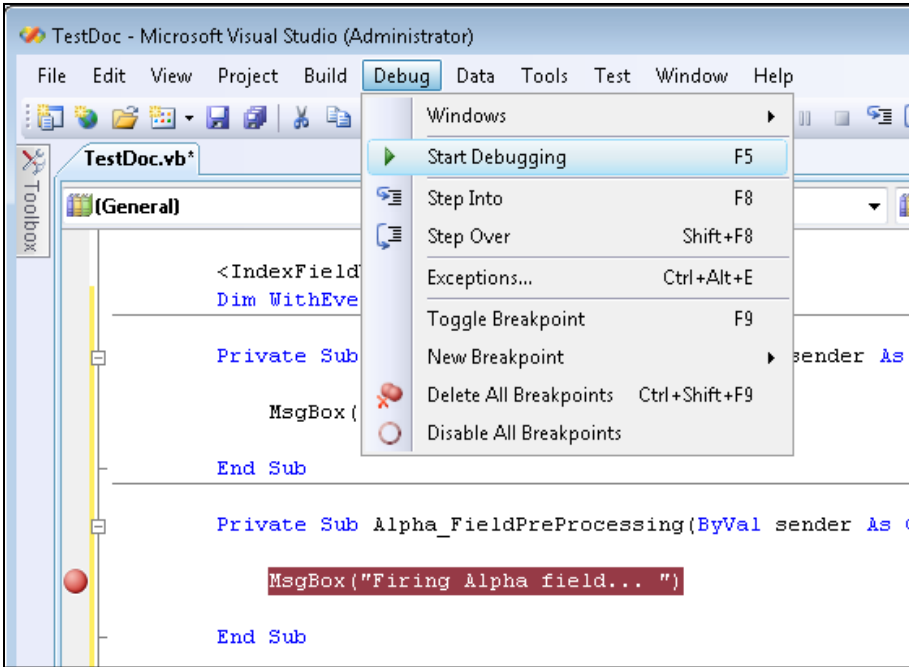


# Debugging Validation (Index) Scripts With Visual Studio .NET

*Application Note*

## Stepping Through the Code

Process a batch through to the Validation module. This can be done by either launching the module directly or by selecting a batch in the Batch Manager and clicking the Process Batch button. Visual Studio .NET is now ready to begin the debugging process and step through the code. Start the Visual Studio debugger from the Debug menu or pressing **F5**.

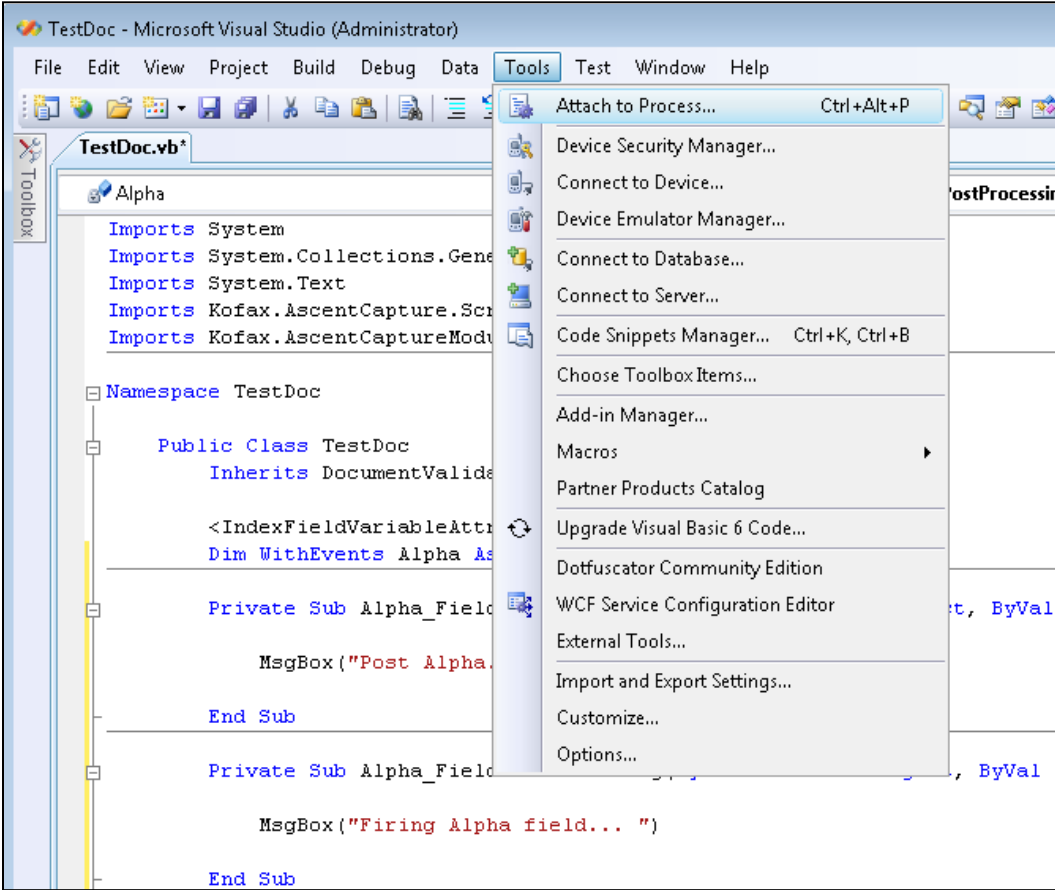


# Debugging Validation (Index) Scripts With Visual Studio .NET

*Application Note*

## Attaching to an Existing Process

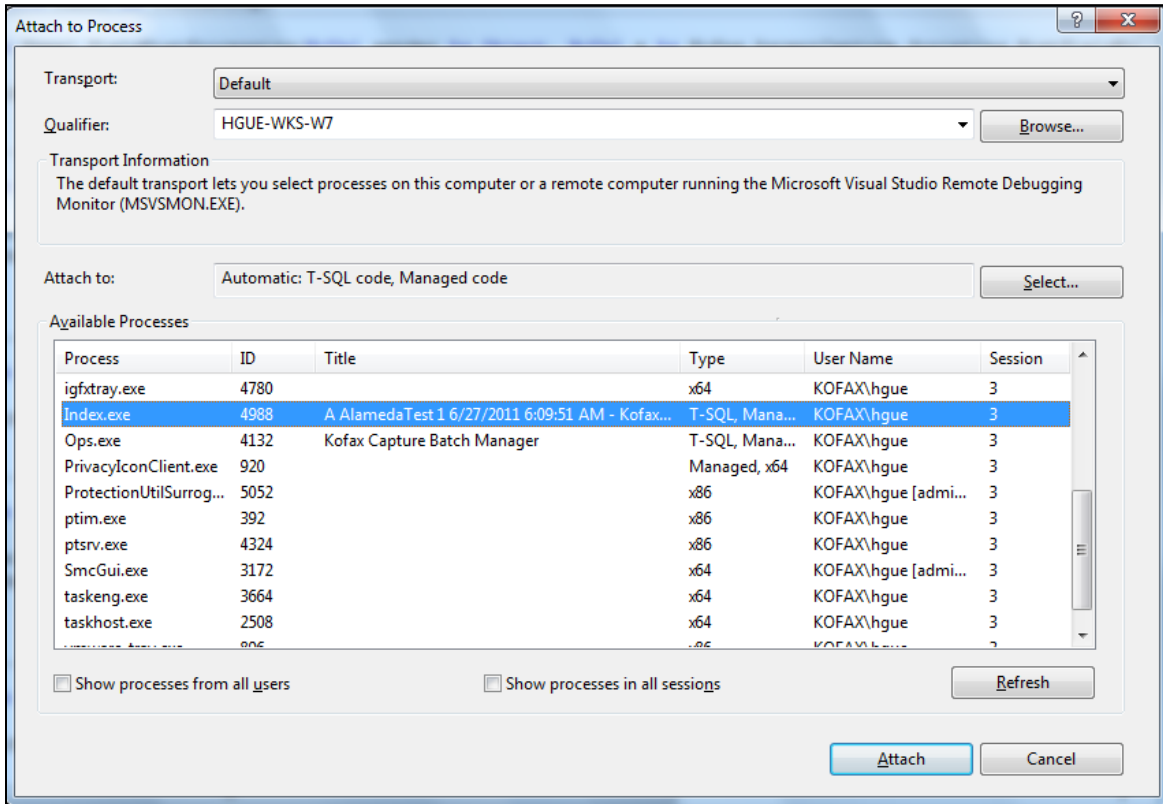
As the project is not set to automatically launch Validation, the project will need to be attached to the process. This can be done from the Processes dialog box. This is available under the Debug menu (Debug | Attach to Process...) and under the Tools menu (Tools | Attach to Process...).



# Debugging Validation (Index) Scripts With Visual Studio .NET

*Application Note*

Select "Index.exe" from the Processes dialog box.



The project will attach to the Validation module and process the document until Validation reaches the selected breakpoint. It will stop and highlight the running line of code.

```

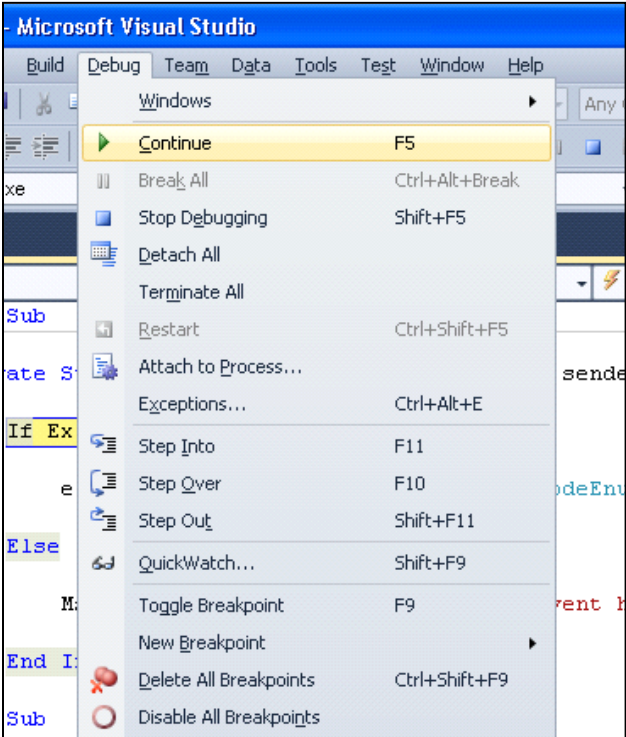
strActionEvent = "Document " & m_nDocumentNumber & " Opened"
Case AscentCaptureModule.KfxOcxEvent.KfxOcxEventDocumentValidating
m_oHeldDocument.CustomStorageString("Kfx.Capture.EarlyExitFlag") = ("Exit")

```

# Debugging Validation (Index) Scripts With Visual Studio .NET

## Application Note

You can now step through the code using "Step Into" **F11**, "Step Over" **F10** and "Step Out" **Shift+F11**. Each time the **F11** key is pressed, one operation will be performed. This will advance to the next line of code and reveal exactly what code is being run. To continue processing normally, simply press the **F5** key, and Export will continue to process until the next time a Breakpoint is encountered or the export process has been completed.



To stop debugging, simply click the "Stop Debugging" **Shift+F5** keys or select the menu entry.

---

## Application Note Summary

Using the techniques noted in this Application Note, a developer can step through a Validation Script as it is running to see the actions taking place and values being used. This will provide the information needed to address any issues such as fixing an error condition that occurs, or modifying the Validation Script to handle any given situation as needed.