

Technical Support

Technical Bulletin

Distribution:	Application Engineers, Technical Support Engineers, Vision Solutions Engineers, Technical Documentation Group, Worldwide Sales	☐ INTERNAL ONLY
Bulletin No.:	TB-08-006	☒ Internal & Customers (as needed)
Date:	November 20, 2008	☐ Internal & Customers (MANDATORY)
Product:	VISIONPRO	Rev: 4.2 to 5.1
Title:	Diagnosing broken data bindings	

Purpose

This document describes a sporadic problem with data bindings that may occur only in VisionPro releases 4.2 through 5.1. As explained in TB-08-005, data bindings are the software objects that move data from a source property of one tool to a destination property of another tool. Data bindings between tools are represented by graphical links within QuickBuild.

This technical bulletin may apply to you if you observe any of the following:

- Data bindings in QuickBuild do not update the destination property when a CogJob or a CogToolGroup runs.
- Properties shown in the user interface of an Application Wizard generated application do not update and reflect old data while the CogJob is running.

This document describes how you can add debugging code to your application to confirm whether it is affected by this data binding problem.

Note: Data binding problems may occur for a wide variety of reasons, especially in programmatic applications. This technical bulletin applies to a very specific issue and does not address all data binding problems. See tech bulletin TB-08-005 for more information about data bindings.

A Brief Explanation of the Problem

In some very rare cases VisionPro tool code will encounter a low-level hardware exception such as a divide-by-zero, or a memory access violation. When this happens, the ToolGroup containing the tool correctly catches the exception and updates the ToolGroup's RunStatus, but the tool itself is left in a subtly broken state: subsequent runs of the tool succeed but the tool never again fires any Changed events. These events are the ones that tell data bindings, or the edit control of the tool, that new output values are available.

Therefore, after a tool experiences a low-level hardware exception it will never again update the destination of its output data bindings. In QuickBuild all tools, including the broken one, will continue to run normally, but the newly computed output values from the broken tool will not be transmitted to downstream tools. In an Application Wizard generated application the output properties of the broken tool will not update: they will permanently display the values they had prior to the hardware exception.

Technical Support

Technical Bulletin

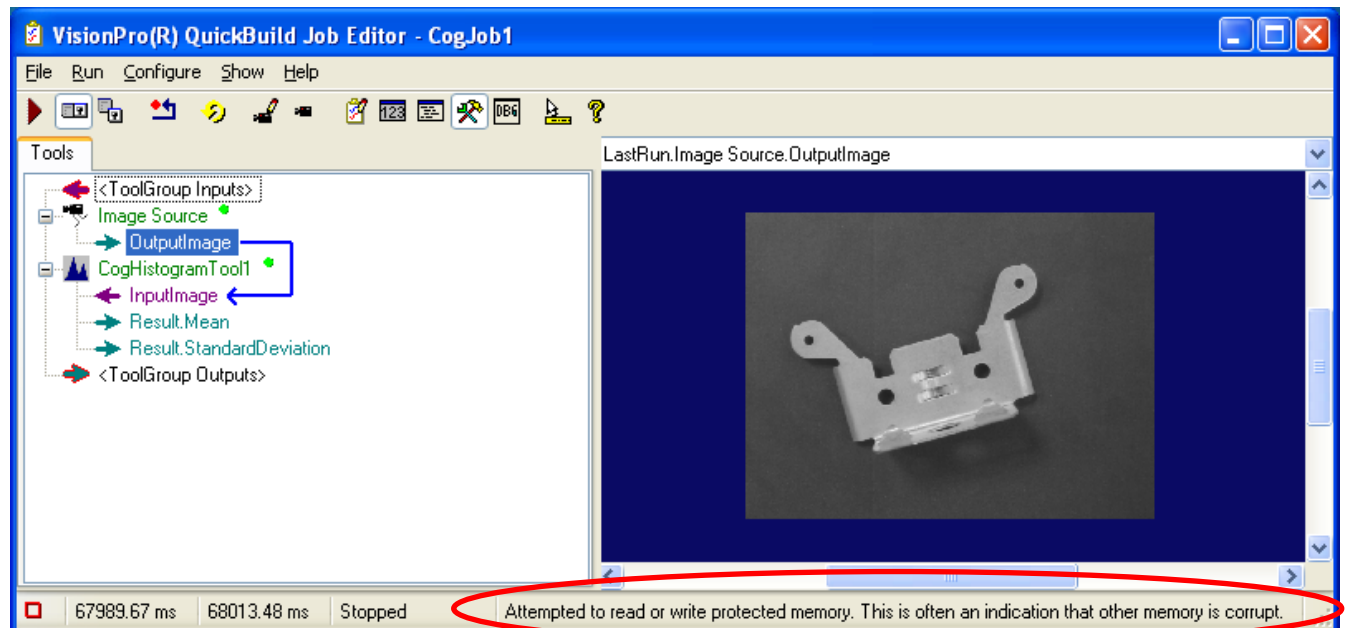
It is important to note that **only** hardware exceptions will cause this problem. A list of the known hardware exceptions can be found here: <http://msdn.microsoft.com/en-us/library/aa363082.aspx>.

Is This Problem Affecting Your Application?

Your application may be affected by this problem if:

- You are using VisionPro release 4.2, 4.3, 4.4, 5.0, or 5.1.
- Your application contains links between tools in QuickBuild that first appear to update for a period of time but then suddenly stop updating.
- If all of the outputs from a single tool have stopped updating.
- The edit control for one of your tools has stopped updating its outputs when the tool is run.

In some cases you may see one of your tools at the exact moment that it experiences a hardware exception. For example, the ToolGroup below has just experienced a memory access violation and is displaying the diagnostic message in the status bar:



In order to help you diagnose this problem, you can add a small amount of script code to your QuickBuild ToolGroups to determine whether this low-level hardware exception is occurring. In the box below is a C# implementation of the GroupRun function that you can use in the QuickBuild script window for your ToolGroup. This script code will **halt** your ToolGroup as soon as a hardware exception has been detected. It will then display a diagnostic message box reporting the type of the exception.

Technical Support

Technical Bulletin

```
// The GroupRun function is called when the tool group is run. The
// implementation provided here is equivalent to the normal behavior of the
// tool group until a hardware exception occurs. When a hardware exception
// occurs the tool group halts and displays a message box.
public override bool GroupRun(ref string message,
                             ref CogToolResultConstants result)
{
    for (Int32 toolIdx = 0; toolIdx < toolGroup.Tools.Count; toolIdx++)
    {
        try          // Run each tool in the tool group using the RunTool function
        {
            toolGroup.RunTool(toolGroup.Tools[toolIdx], ref message, ref result);
        }
        catch(System.Exception e)
        {
            // If this is some type of hardware exception, halt and display a
            // message box. You can expand this if statement to contain more
            // types of hardware exception, but these two are the most frequent..
            if (e is System.AccessViolationException
                || e is System.DivideByZeroException)
                MessageBox.Show(e.GetType().Name + ": " + e.Message);

            throw;      // Re-throw the exception so it is handled normally.
        }
    }

    // Returning False indicates we ran the tools in script, and they should
    // not be run by VisionPro
    return false;
}
```

How to Remedy the Problem

If, after updating your application with the diagnostic code above, you detect one of these hardware exceptions we recommend that you upgrade to software release VisionPro 5.2 or later. This upgrade will correct the problem: hardware exceptions will be caught properly by VisionPro software without disabling all future Changed events.

If upgrading your version of VisionPro is not an option you may be able to add code to your application that catches hardware exceptions from each tool's Run() function and then completely re-creates the just-broken tool. Implementing this strategy requires a non-trivial amount of effort and should only be attempted if (a) you are sure that you are being affected by the problem described above, and (b) you absolutely cannot upgrade to VisionPro 5.2 or later.