

Technical Support

Technical Bulletin

Distribution:	Application Engineers, Technical Support Engineers, Vision Solutions Engineers, Technical Documentation Group, Worldwide Sales	☐ INTERNAL ONLY
Bulletin No.:	TB-07-001	☒ Internal & Customers (as needed)
Date:	September 15, 2007	☐ Internal & Customers (MANDATORY)
Product:	VISIONPRO	Rev: ALL

Title:	"System Out of Memory" Exception	
---------------	----------------------------------	--

PURPOSE

This article describes a "System Out of Memory" exception that may occur in a VisionPro application and provides tips and guidelines for minimizing its occurrence.

TECHNICAL DETAILS

There is a limitation on all Microsoft 32-bit operating systems (32-bit Editions of Windows 2000, XP, Vista) that each application is only allowed [2 gigabytes \(GB\) of virtual address space](#). If a VisionPro application attempts to exceed – or even to approach – the 2 GB limit a "System Out of Memory" exception will occur.

The 2 GB limit is a limit on the size of the *virtual* address space. The Windows operating system manages the virtual address space, usually splitting it between two physical storage locations: physical memory and a page file on the hard disk. The 2 GB limit cannot be increased by adding more physical memory, or by increasing the size of the page file. Even if your PC has 8 GB of RAM *and* a 32 GB page file your application is still limited to 2 GB of address space.

Due to normal fragmentation of the memory space, the amount of memory available to a real-life application may be significantly less than 2 GB. This is especially true in VisionPro applications where large images are in use.

The exact amount of memory required for a VisionPro application is difficult to compute. The memory usage depends upon many factors including:

- ?? The images: how many are there, and how big are they?
- ?? The vision tools: which tools are used, how many are there, how are they configured?
- ?? The application parameters: how many Jobs are used, how many cameras, how many items are being stored in the PostedItems queue?

All of these play an important role in determining overall memory usage. The "System Out of Memory" exception can occur when using small or large image sizes or when using a small or large number of vision tools.

If you are observing a "System Out of Memory" exception there are things that you can do within QuickBuild and your application to help reduce memory usage. In some applications, a "System Out of Memory" is not thrown and instead you may observe an "Acquisition Error: acquisition failed abnormally [Error code #]".

If you are observing either of these messages the following features (mostly in QuickBuild) can be used to help control memory usage in large memory applications.

Technical Support

Technical Bulletin

?? **Posted Items** The Posted Items queue within QuickBuild is used for storing result information. It can store images, tool records, or any other properties that you wish to make available. By default the Posted Items queue stores the LastRunRecords generated by the running Jobs. It stores the 32 most recently generated records.

In a finished application (e.g. an application produced by the Cognex Application Wizard) items are removed from the queue and presented to the user. In QuickBuild the items are simply stored for future inspection. The presence of these items increases the QuickBuild memory usage. There are two approaches to reducing the memory usage: reduce the number of items in the queue and/or reduce the amount of stuff in each item. Both approaches are presented below.

- o **Queue Size:** This parameter controls the number of items in the queue. Set it to a smaller number to reduce memory usage. To modify this parameter, select “Configure QuickBuild Application Properties...” from the QuickBuild menu and adjust the size as shown in Figure 1. To see the items in the Posted Items queue, select “Show Floating Results...” from the QuickBuild menu.

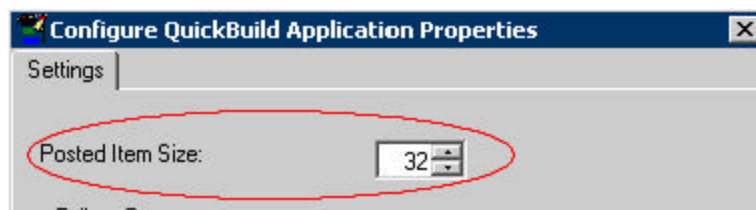
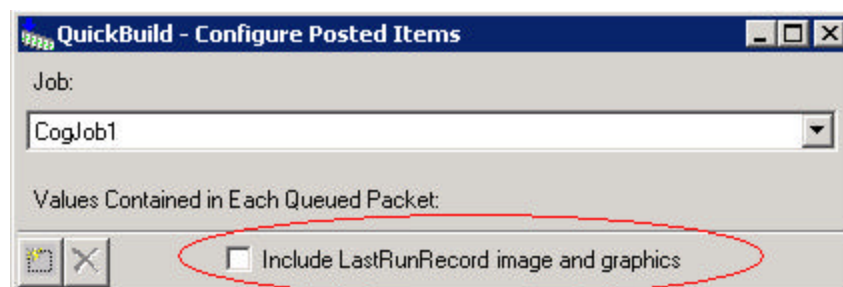


Figure 1. Adjusting the Posted Items Queue Size

- o **Queue contents:** By default each job will add its LastRunRecord to the queue. If you are using lots of tools, or large images, this record may be undesirably large. You can adjust the contents of the queue by selecting “Configure Posted Items...” from the QuickBuild menu. Clear the “**Include LastRunRecord image and graphics**” check box (as shown in Figure 2) to remove the LastRunRecord entirely. Do this for each job if you wish to eliminate all LastRunRecords from the queue.



Technical Support

Technical Bulletin

Figure 2. Removing last run tool record

?? **Vision Tool Result Graphics:** Each job produces a LastRunRecord – even if that record isn't stored in the Posted Items queue. The contents of this record can be controlled, in part, by the choices you make for each tool's result graphics. Some vision tools, like PatInspect and Blob, can be configured to produce additional images in the LastRunRecord. By turning off these images you can decrease the size of the LastRunRecord. You can turn off unwanted images using the graphics tab on each tool's edit control. For example, Figure 3 shows how to disable the result images from PatInspect, and Figure 4 shows how to disable the diagnostic output images from Blob. To minimize memory usage you should also ensure that the option "Show Input Image without copy" is selected.

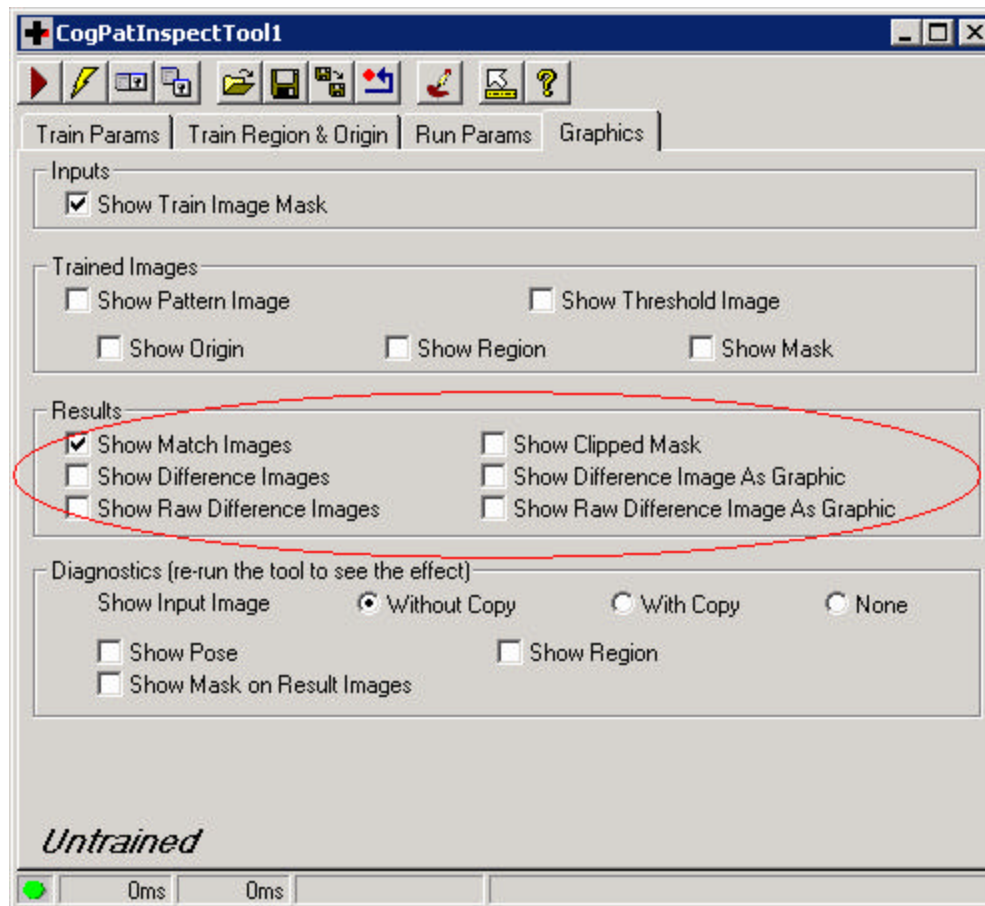


Figure 3. Adjusting PatInspect tool result output images

Technical Support

Technical Bulletin

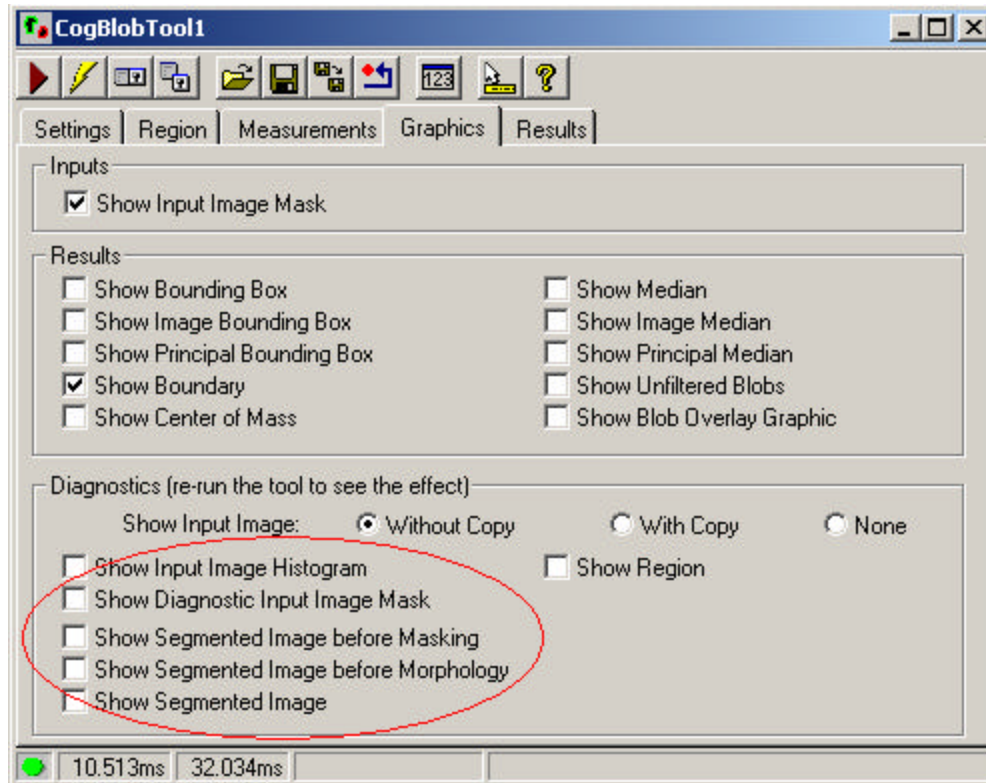


Figure 4. Adjusting Blob output result images

?? **Acquisition and Large Images:** Using large input images, such as those generated by line scan cameras or color cameras, will result in larger memory usage. This effect is compounded when using multiple cameras or multiple Jobs.

If you are observing a message that says: "Acquisition Error: acquisition failed abnormally [Error code #]", where # is a value between 1 and 20, then try adjusting the amount of memory allocated for acquisition on the MVS-8600/MVS-8600e frame grabber.

- o **MVS-8600/MVS-8600e:** You can control the amount of system memory allocated for acquisition if you are using a MVS-8600 or MVS 8600e Camera Link frame grabber.

Starting in VisionPro 4.4 (or later) the memory amount can be controlled programmatically using the *ICogAcqHardwareImagePool* interface, or it can be controlled in QuickBuild as shown in Figure 5. You can select "Use Custom Value" and specify the number of images to allocate in the "Image Source" form, under the Image tab.

Technical Support

Technical Bulletin

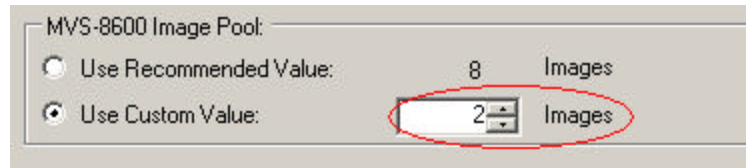


Figure 5. Adjusting the number of images for Camera Link acquisition

Prior to VisionPro 4.4, the ICogAcqHardwareImagePool interface did not exist and acquisition memory used for storing MVS-8600 Camera Link frame grabber images can be adjusted using the registry key:
`HKLM\System\CurrentControlSet\Services\fg8600\PRVPH\MaxCPUBuffer`. The value is expressed in hex, and indicates the number of megabytes of memory to reserve for the acquisition system **for each** acquisition fifo. The default value is 0x20, which corresponds to 32 megabytes of system memory. Increase this value to a number that accommodates the frame size desired. A good starting point is to specify no less than four times the number of megabytes required for one frame acquired by one camera.

Special note that in VisionPro 4.2 and earlier the hex value was not applied on a per fifo basis, but was applied to the all acquisition fifos for each board.

- ?? **Training Region:** Some Vision Tools like PMAlign and PatInspect require training images. The size of the training region is controlled on the Train Region and Origin tab of the tool's edit control. Whenever possible you should train using a reduced region of the image, rather than training from the entire image. This can greatly reduce the amount of memory used by the tool. See Figure 6 on changing the Train Region shape.

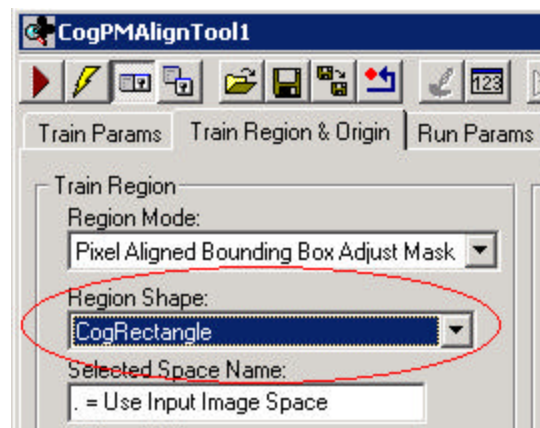


Figure 6. Adjusting the train region to a smaller shape

- ?? **Failure Queue:** The Failure queue within QuickBuild is used to store "failed" images. The size and contents of this queue can be controlled from the "Configure QuickBuild Application

Technical Support

Technical Bulletin

Properties” form as shown in Figure 7. Reducing the number of failed images that can be stored may reduce the amount of memory in use. The Failure Logging mechanism can also be turned off entirely by enabling the checkbox labeled “Accept”.

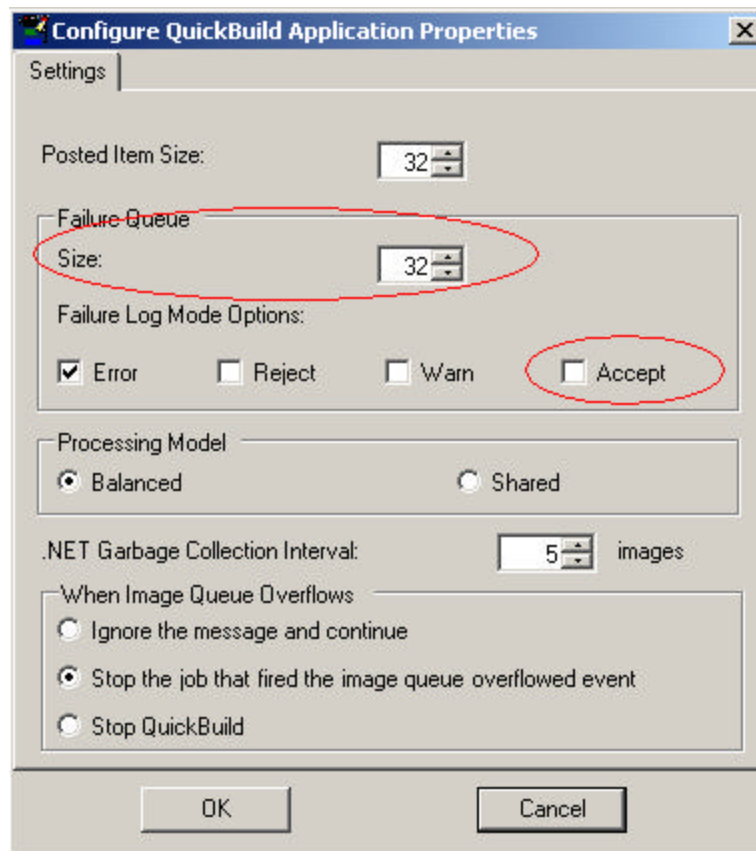


Figure 7. Adjusting the Failure Queue Size

- ?? **Display Windows:** Closing display windows can reduce the amount of memory and potentially free up memory when not visible. Minimizing display windows may not necessarily free up as much memory as closing the window. The recommendation is to close as many forms as possible and close the display window inside of forms. On the Job Form menu bar the Show menu item has options to close the local display and the floating display. Displays inside of applications should also be closed. Images referenced or added to a display will not be freed until the display window itself is destroyed.
- ?? **Garbage Collection:** The .NET garbage collection interval can be control from the “Configure QuickBuild Application Properties” form as shown in Figure 8. Reducing the .NET garbage collection interval may help reduce memory usage after each Job is run, but may not necessarily reduce the overall memory usage for the application.

Technical Support

Technical Bulletin

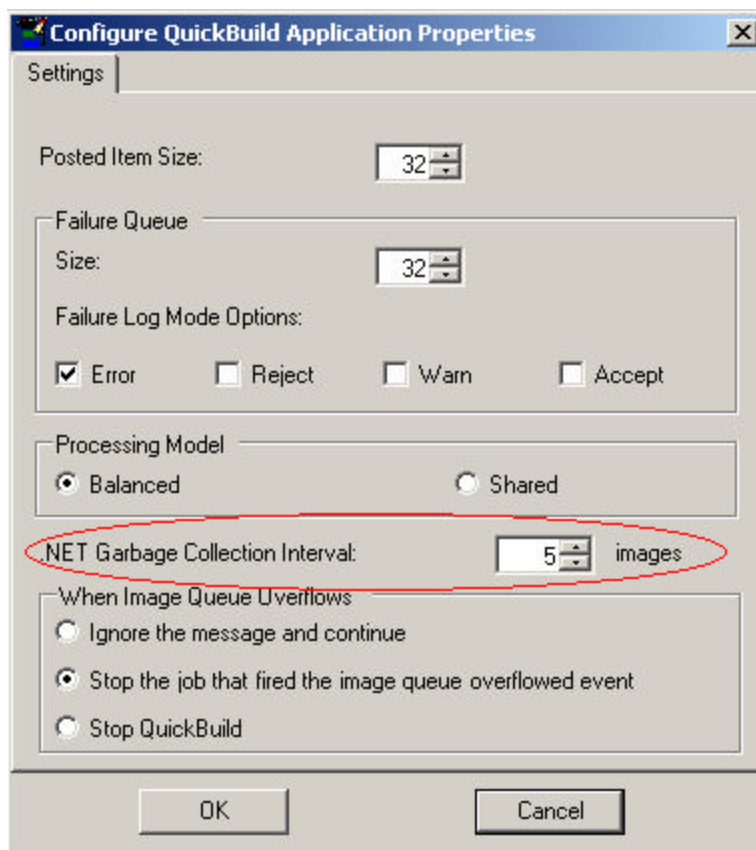


Figure 8. Adjusting the .NET Garbage Collection interval

- ?? **Application Wizard:** The application generated by the Application Wizard contains queues that hold onto tool records. These queues are used to show the scrolling pass/fail records along the top of the generated application. There are two queues for each job. You can reduce the size of the queues by modifying the source code that the Application Wizard generates. The following global variables in the generated code may be reduced, but not less than 1.

```
private const int AllQueueLength = 16;    // default
private const int FailureQueueLength = 16; // default
```

MOVING FORWARD

Following the guidelines outlined in this article should help reduce the amount of memory used by your VisionPro application. You can monitor the amount of memory used in your application using the Microsoft Task Manager.