Topics

Compatibility

1. Supported platforms and system requirements

Which versions of the Microsoft .NET Framework* does the VTune™ Performance Analyzer 7.1 support?

IA-32 systems support versions 1.0 and 1.1 of the .NET Framework*. The VTune analyzer does not support profiling for the .NET Framework* on Itanium®-based systems.

2. Profiling Managed Extensions in C++ programs

Can the VTune analyzer profile Managed Extensions in C++ programs?

Yes. This includes .NET* DLLs, native IA-32 DLLs, and mixed DLLs that contain both managed and unmanaged code.

3. Unpacking into an existing project directory

I cannot unpack a project into an existing project directory?

The VTune analyzer can't unpack a project (associated with pre-jitted files) into an existing project directory. So you may face this problem if you profiled a .NET* program with a few pre-jitted .NET* modules, or .NET* system modules. The work around is to unpack into a new project directory.

4. Optimizations lost during profiling

My .NET* application seems to lose all optimizations when being profiled?

If you are profiling your application using .NET Framework* version 1.0.3705, this is probably true. To get most optimizations back, you need to create an INI file with the same base name as the executable — for example, create ***app-name*.ini** in the same directory where your executable, ***app-name*.exe**, resides. The file should contain the following two lines:

```
[.NET Framework Debugging Control]

AllowOptimize=1
```

Usage

5. Not viewing unmanaged DLLs executed under the profiled process

How do I get a call graph of my .NET* application only, without all the native DLLs that execute under the profiled process?

Use the call graph wizard, and select **.NET Profiling**. Specify the relevant information about your application in the various wizard windows. The last window allows you to choose between call graph with .NET classes only, and a complete call graph that includes both .NET and native calls. The default is **.NET classes only**.

6. Viewing unmanaged DLLs executed under the profiled process

How do I get a call graph of my .NET* program including all the native DLLs that execute under the profiled process?

Use the call graph wizard, and select **.NET Profiling**. Specify all the relevant information about your application in the various wizard steps. The last step enables you to choose between call graph with .NET classes only, and a complete call graph that includes both .NET and native calls. The default is **.NET classes only**. Select **All .NET classes and all native modules** to get both .NET and native code call graph data.

7. Profiling ASP.NET* applications

How can I profile my ASP.NET* application?

The sampling and call graph wizards guide you through the various stages required to profile an ASP.NET* application.

It is recommended that you shut down all clients of all Internet Services before invoking an ASP.NET* profiling session, and before stopping the profiling session. The VTune analyzer restarts IIS as part of the profiling startup and termination process. This might have unpredictable effects on client applications, including system crashes.

8. Profiling COM+ .NET* services

How can I profile my COM+ .NET* service?

The sampling and call graph wizards guide you through the various stages required to profile a COM+ .NET* service. You need to execute your COM+ .NET* service on the machine where you intend to profile it, at least once before profiling. This will ensure all COM+ services are properly registered and displayed for selection in the call graph wizard.

See the next item for additional information on profiling COM+ .NET* services on Windows* Server 2003*.

Please refer to the COM+ profiling white paper at
http://www.intel.com/software/products/vtune/vpa/callgraph.pdf for more information.

9. Profiling COM+ .NET* services on Windows* Server 2003*

How can I profile my COM+ .NET* service on Windows* Server 2003?

Call Graph profiling of COM+ .NET* services is not supported on the Windows* Server 2003*.

For sampling you need to change the identity to **This User**, and set the user name and password. Follow these steps to get this done:

1. Open **Control Panel** → **Administrative Tools** → **Component Services**.
2. In the **Component Services** dialog, on the left side, open **Component Services** → **Computers** → **My Computer** → **COM+ Applications**.
3. Select your COM+ .NET* service.
4. Right-click on it and select **Properties**.
5. Go to the Identity tab. **Interactive user** should be selected by default.
6. Change the account to **This user** and input the user name and password into the corresponding fields.
7. Click **Apply** to save changes.

10. Profiling .NET* programs invoked via a script

How can I profile a .NET* program invoked via a script?

The call graph and sampling wizards enable you to specify a script that invokes your application.

Tips

11. Missing functions in Activity results

How come I don't see specific functions in Sampling or Call graph Views; while I'm sure these functions consume a lot of CPU cycles in my application?

The .NET* JIT Compiler sometimes inlines functions into their calling functions. As a result, the CPU consumed by the callees is associated with the callers, and there is no explicit mention of the callees.

12. Launching applications outside of the VTune analyzer

Can I profile a .NET* application that I launch outside of the VTune analyzer?

Yes. Please refer to the VTune analyzer online Help for more details.

13. Systems executing slowly after profiling an ASP.NET* application

After profiling an ASP.NET* application that crashed during profiling, my system seems to execute slower than usual. Is it somehow related to the earlier crash?

Your system might still be set for profiling .NET* applications. Please check the system environment variables. If there is a variable named **Cor_Enable_Profiling**, remove it and restart any .NET* application.

14. COM+ .NET* service not visible in wizard

Why do I not see my COM+ .NET* service in the list of available services when configuring the call graph collector using the VTune analyzer wizard?

Your service might register itself dynamically, so run your application directly, not via the VTune analyzer, to ensure proper registration. Thereafter, you should see your service in the list of COM+ Services displayed by the VTune analyzer wizard.

15. Using Pause/Resume API in .NET applications

Why is my .NET application that uses Pause/Resume API not able to find CLRVTuneAPI assembly?

This may be because **CLRVTuneAPI** has not been registered in the .NET Global Assembly Cache. The problem can be resolved by manually registering this assembly. To do this, invoke a command window and change to the VTune analyzer's Analyzer\Bin directory. At the command prompt, type in the following command:

```
gacutil –i CLRVTuneAPI
```

The **gacutil** utility is usually located in the .NET Framework install directory. You will need to specify the gacutil utility's full path, if it is not included in your **PATH** environment variable.