

Tools - DotNET - DotNET analysis checklist for troubleshooting any crashes

Purpose

This page talks about the prerequisites and checklist to be followed before doing the .NET analysis and also to troubleshoot any crashes while performing .net analysis.

Please select the section you are interested in from the list below.

- [.NET Analyzer Pre-requisite](#)
- [Assembly reference and Project reference](#)
 - [Packaging of the .NET assemblies](#)
- [C#.Net](#)
 - [Analysis Issues](#)
 - [Code Analysis](#)
 - [Memory Issues](#)
- [VB .Net](#)
 - [Analysis Issues](#)
 - [Code Analysis](#)
 - [Memory Issues](#)

Applicable in CAST Version

Release	Yes/No
8.3.x	<input checked="" type="checkbox"/>

Applicable RDBMS

RDBMS	Yes/No
CSS	<input checked="" type="checkbox"/>

.NET Analyzer Pre-requisite

Assembly reference and Project reference

In the application if the dependencies between projects has assemblies, then we need to look in the application. If we have the corresponding application extension (dll) in the defined folder of the project file if it is not present then we need to ask for the dll from the customer/application team and then do the packaging again after getting the dll, packaging which can be done as described below:

- Put the dll in the path mentioned in the project file.
- Put all the dlls in one folder and add that folder as a .net dll package.

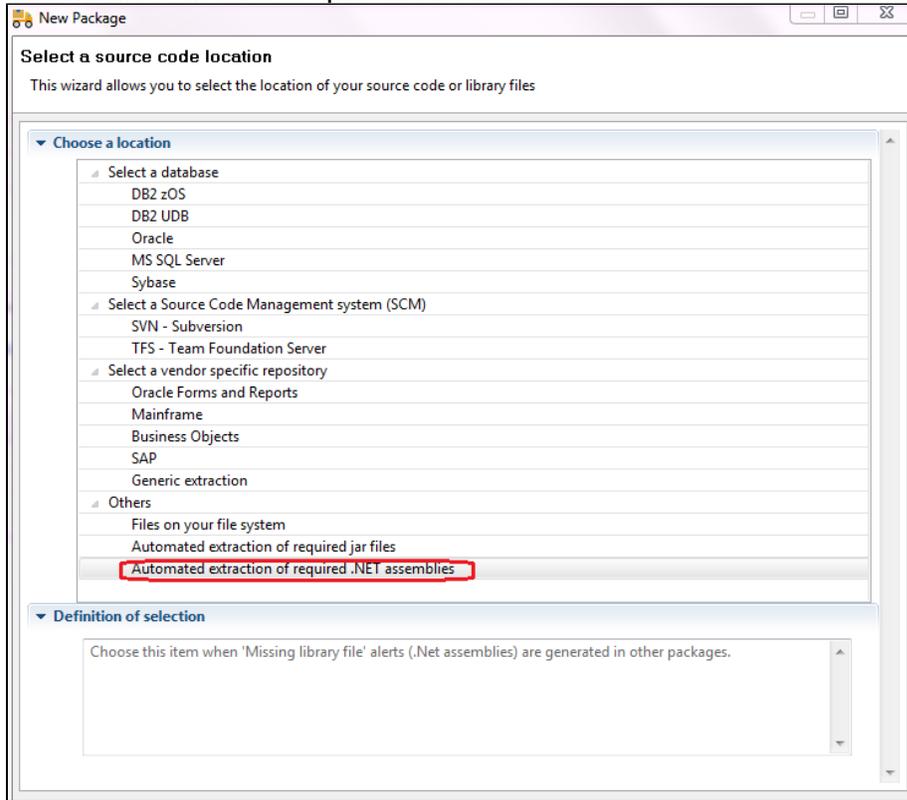
If the dependencies between the projects has another project then we need to look in the application for the presence of that project file which should ideally be present.

Packaging of the .NET assemblies

The .net assemblies need to be packaged along with the source code before performing the .net analysis.

The package should be done as follows:

1. Open DMT.
2. Add new package
3. Select **Automated extraction of required .net assemblies**

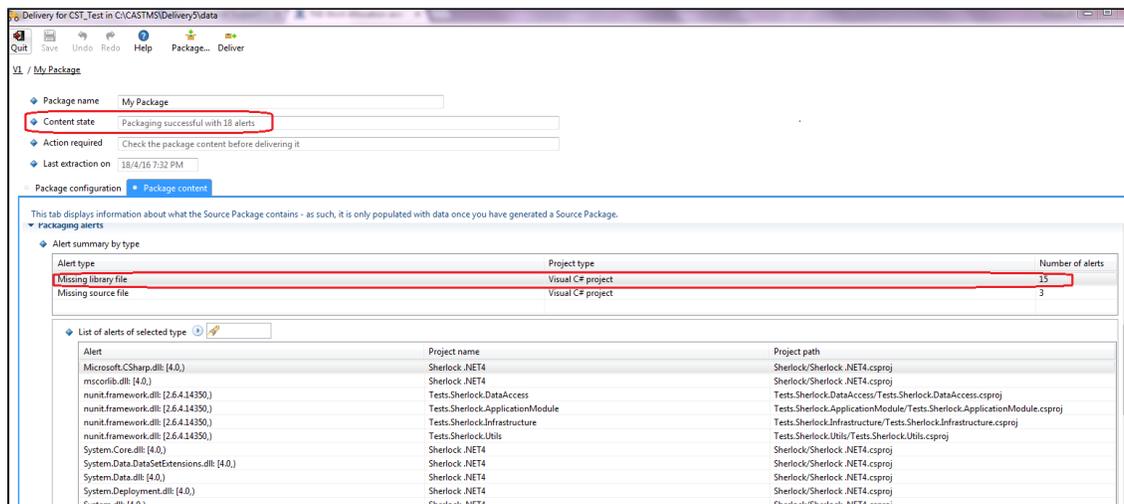


4. Defining a package on `C:\Windows\assembly` does not extract the system assemblies. Use the below instructions instead.
 - a. Choose `C:\Windows\Microsoft.NET\Framework\v4.0.30319`. Choosing between 32 & 64 bits is mostly a non issue. What matters is the interface, and the interface is the same in both cases. But it is important to choose a folder where you will have only one version of system libraries. If you don't, then for some projects a version might be selected, for other projects, another version, which may lead to ambiguities later.
 - i. So, if you know none of the project requires a version more recent than xxx, just select `C:\Windows\Microsoft.NET\Framework\v"xxx"`
 - ii. If you don't know (or don't care), just select: `C:\Windows\Microsoft.NET\Framework\v"latest available"` Which will most of the time be: `C:\Windows\Microsoft.NET\Framework\v4.0.30319`
5. Package it
6. Check the required dll of other version need to be remediated, follow the below page which also explains how to remediate the alerts: [Delivery Manager Tool - Information - Project Remediation - Remediate packaging alerts](#)

Find an example below which explains how to remediate if you have missing assemblies, or if you have missing source files.

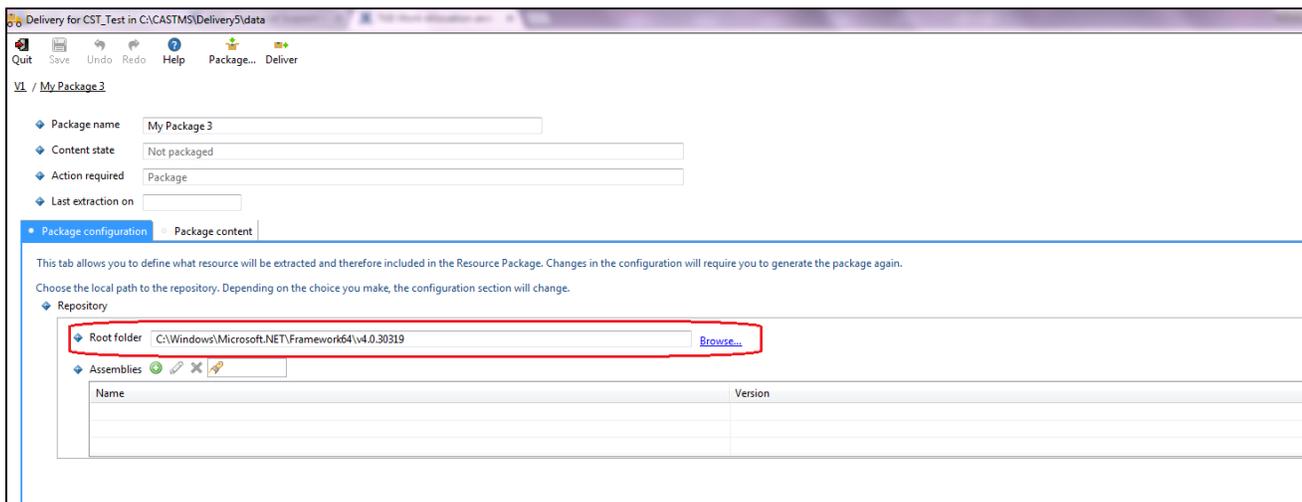
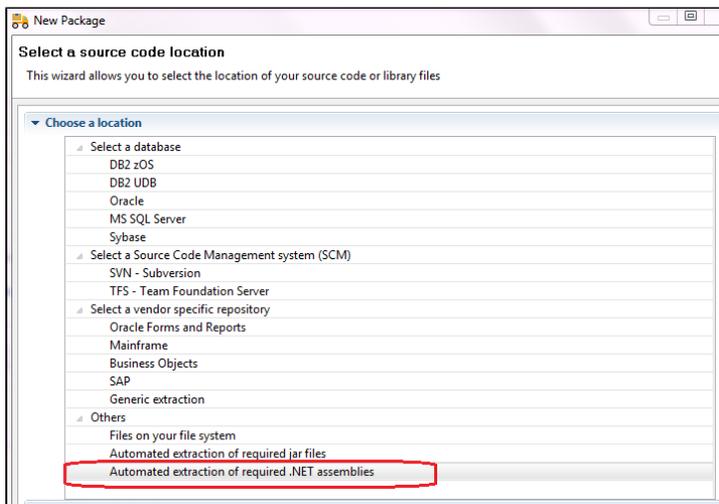
In the below screen shot we have 18 alerts related to missing assemblies and source files.

We have the missing assemblies mentioned below under the package content



Since the missing assemblies if of .Net framework version 4, so we need to package the .Net assmeblies of framework 4.

In DMT we have to select the automated extraction of required .Net assemblies.



After packaging we can see that in the .NET assemblies package, in the package configuration tab, there is the list of all the alerts detected, and in the package content tab the list of assemblies that found to remediate these alerts.

Package configuration | Package content

This tab allows you to define what resource will be extracted and therefore included in the Resource Package. Changes in the configuration will require you to generate the package again.

Choose the local path to the repository. Depending on the choice you make, the configuration section will change.

Repository

Root folder: C:\Windows\Microsoft.NET\Framework64\v4.0.30319

Assemblies

Name	Version
Microsoft.VisualBasic.dll	(10.0.)
microsoftlib.dll	(4.0.)
System.Core.dll	(4.0.)
System.Data.DataSetExtensions.dll	(4.0.)
System.Data.dll	(4.0.)
System.Deployment.dll	(4.0.)
System.dll	(4.0.)
System.Drawing.dll	(4.0.)
System.Windows.Forms.dll	(4.0.)

Package configuration | Package content

This tab displays information about what the Source Package contains - as such, it is only populated with data once you have generated a Source Package.

Projects found

Number of Added/Removed items is based on the cloned version of this package

Project summary by type

Project type	Technology	# Selected	# Ignored	# Added	# Removed
Net Resources	CF	11	0	0	0

List of projects for the selected type

T	Name	Path	Selected/ignored	Compared with last version
	Microsoft.VisualBasic.dll		Selected	N/A
	microsoftlib.dll		Selected	N/A
	System.Core.dll		Selected	N/A
	System.Data.DataSetExtensions.dll		Selected	N/A
	System.Data.dll		Selected	N/A
	System.Deployment.dll		Selected	N/A
	System.dll		Selected	N/A
	System.Drawing.dll		Selected	N/A
	System.Windows.Forms.dll		Selected	N/A
	System.Xml.dll		Selected	N/A
	System.Xml.Linq.dll		Selected	N/A

Then, after packaging we see that there are no more alerts related to missing assemblies in the DMT.

Now, in the below screen shot you could see there is a third party assembly which is missing.

Delivery for CST_Test in C:\CASTMS\Delivery\data

VI / My Package

Package name: My Package

Content state: Packaging successful with 7 alerts

Action required: Check the package content before delivering it

Last extraction on: 18/4/16 7:32 PM

Package configuration | Package content

This tab displays information about what the Source Package contains - as such, it is only populated with data once you have generated a Source Package.

Name	Path	Selected	Compared with last version
Tests.Sherlock.Infrastructure	Tests.Sherlock.Infrastructure/Tests.Sherlock.Infrastructure.csproj	Selected	N/A
Tests.Sherlock.Utils	Tests.Sherlock.Utils/Tests.Sherlock.Utils.csproj	Selected	N/A
Tests.Utils	Tests.Utils/Tests.Utils.csproj	Selected	N/A

Packaging alerts

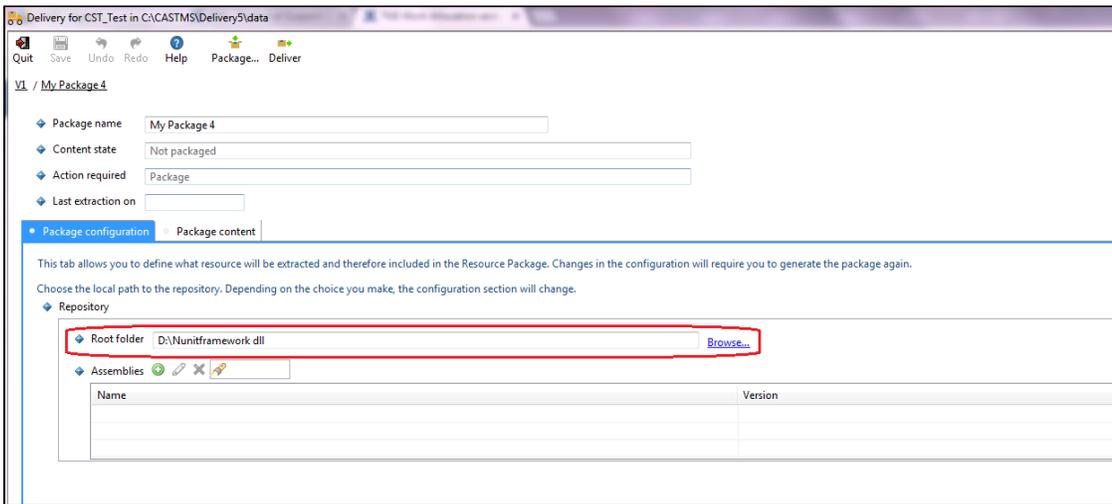
Alert summary by type

Alert type	Project type	Number of alerts
Missing library file	Visual C# project	4
Missing source file	Visual C# project	3

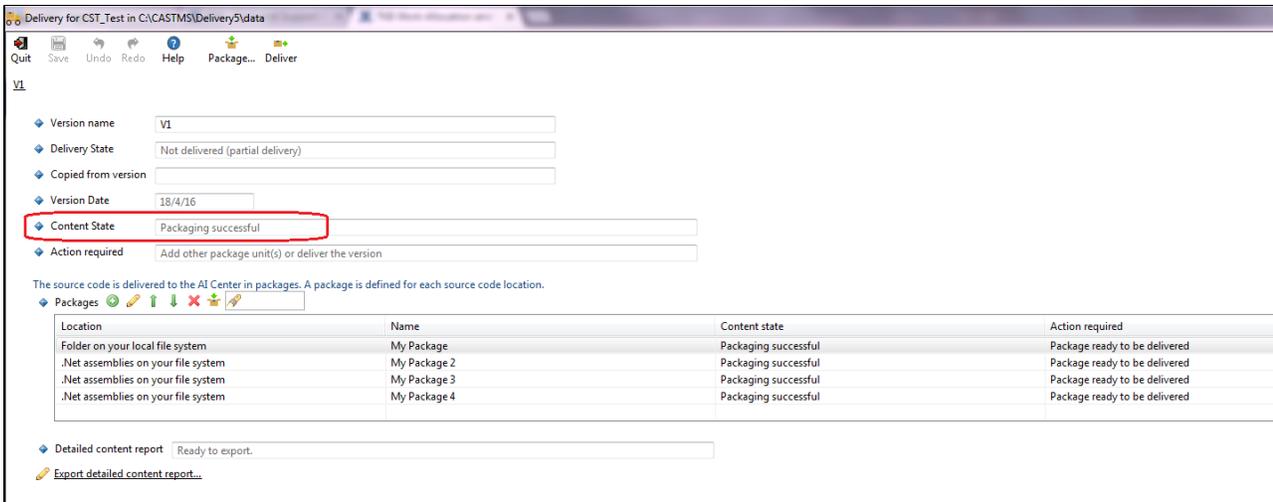
List of alerts of selected type

Alert	Project name	Project path
nunit.framework.dll: [2.6.4.14350.]	Tests.Sherlock.DataAccess	Tests.Sherlock.DataAccess/Tests.Sherlock.DataAccess.csproj
nunit.framework.dll: [2.6.4.14350.]	Tests.Sherlock.ApplicationModule	Tests.Sherlock.ApplicationModule/Tests.Sherlock.ApplicationModule.csproj
nunit.framework.dll: [2.6.4.14350.]	Tests.Sherlock.Infrastructure	Tests.Sherlock.Infrastructure/Tests.Sherlock.Infrastructure.csproj
nunit.framework.dll: [2.6.4.14350.]	Tests.Sherlock.Utils	Tests.Sherlock.Utils/Tests.Sherlock.Utils.csproj

We need to create another package and provide the path of the third party assembly.



After packaging the third party assembly the package is fine with no more DMT alerts, please find the screen shot below.



Following is the list of the main problems that can occur during a .NET analysis using the .Net analyzer. There are several types of problems: related to not following the prerequisites of the analysis, related to memory consumption or related to the code analysis.

C#.Net

Analysis Issues

In some cases, it is possible to encounter issues related to the analysis prerequisites. In the table below you will find the main messages that may be displayed by the Analyzer. In most cases the issue can be resolved by applying the suggestion in the column **Resolution**.

We need to follow the prerequisites properly to avoid the errors.

Example	Description	Resolution
Unable to locate the .Net Framework path. The product version of this project is PROJECT_VERSION.	The .NET Framework required for the version of the application is not available	Please install the required .Net Framework.
Unable to find the technology LANGUAGE for selected projects.	The technology required for the language of the projects is not available.	Need to check what all languages customer is analyzing. Check if customer is using any language which is not supported by CAST.
No valid input paths specified. Please specify at least one existing project (or website) path on the command line.	There is no project or website selected.	Select at least a project or a website.
None of the selected projects (and/or websites) has its target framework installed. In order to run the analysis, you will need to install the .Net frameworks targeted by the selected projects (and/or websites).	The required .Net Frameworks are not available.	Follow the below page: CMS Snapshot Analysis - Run Analyzer - Non Fatal Error - DotNet - troubleshooting - None of the selected projects has its target framework installed
Bad application configuration because the following file(s) FILES is (are) missing; reinstalling the application might fix the problem.	Some of the application files are missing.	Check which application file is missing from the analysis log
Unable to find any .Net Framework. In order to run the analysis, you will need to install all the .NET framework versions used by the application. To check the installed .net framework, follow this page : How to check if Dot net Framework is installed	The required .Net Frameworks are not available.	If none of the framework is not installed we get this error in the log, so we need to ask to install the correct framework. CMS Snapshot Analysis - Run Analyzer - Non Fatal Error - DotNet - troubleshooting - None of the selected projects has its target framework installed
Failed to load metadata dispenser (CODE). Possible reasons are: missing .NET framework 2.0 version or .NET framework 2.0 has been installed after a more recent version.	The metadata dispenser is not available.	This issue comes because of .Net framework not installed at the customer environment. So we need to check the correct target framework and then ask the customer to install the same. CMS Snapshot Analysis - Run Analyzer - Fatal Error - DOTNet with Core CAST AIP - Failed to load metadata dispenser

Code Analysis

During the analysis, issues can appear related to the code. In the table below you will find the main messages that may be displayed by the analyzer:

Example	Resolution
Analyzer internal error.	There could be some file which might not allow the analyzer to proceed further with the analysis, so we need to follow the following page: CMS Snapshot Analysis - Information - Overview of the Dichotomy process
While analyzing the source code analyzer failed to parse XML file	CMS Snapshot Analysis - Run Analyzer - Non Fatal Error - DOTNet with Core CAST AIP - Failed to parse xml file type http w3 org not found
While computing conversions between an expression and a type, the type of the expression is unexpectedly null.	This is the code issue and need to consult the customer to check the code.
Analyzer internal error or performance issue	In the source code if we have 0kb or file size greater than 2 MB then we need to remove those files and run the analysis

In other case get the source code from customer and reproduce the issue, report to R&D. This page provides the [guidelines on reproducing the issue](#).

Memory Issues

During the analysis, some issues related to memory consumption can occur.

To confirm the memory issue we need to activate the AMT profiling and run the analysis, if CAST rest bytes in more than 1.8GB then it is a memory issue and then we need to follow the below troubleshooting page:

CMS Snapshot Analysis - Run Analyzer - Fatal Error - Analyzer is running out of memory - This may occur under disk full condition

Shown below are the main error messages that may be displayed by the analyzer:

Example	Description
Invalid null pointer argument.	Fatal error.
MemMngrDLL_AllocBlock(): Cannot allocate a new user area of bytes. A memory allocation call failed while creating a node.	The analysis will end because of out of memory.
MngTreeDLL_MkTreeElement() - Not enough memory for building a new 'NODE'.	
Analysis is running out of memory - This may occur under disk full condition	The analysis will end because of out of memory.

VB .Net

Analysis Issues

Example	Description	Resolution
Unable to instantiate component NetReflect ('HRESULT').	Netreflect component not registered in analysis machine registry	Follow the below page CMS Snapshot Analysis - Run Analyzer - Non Fatal Error - DOTNet with Core CAST AIP - Unable to instantiate component NetReflect - XXXXXXXX
AspNet compilation failure could not find generated assembly, please make sure the page compiles	This occurs in the web project / websites when the code is not compiled before the analysis. Compilation of ASP.NET pages is a tricky process in 7.3.X, not related to CAST but related to the fact that to analyze MSIL code of ASP.NET pages we need to execute each ASP.NET page before doing MSIL Analysis using legacy .Net Analyzer. New VB.NET Analyzer in 8.0 doesn't need to compile or execute ASP.NET pages, they are just analyzed by looking at source code. This was one of the main objectives in 8.0 to remove any ASP.NET compilation for VB.NET source code.	Follow the below page: CMS Snapshot Analysis - Run Analyzer - Non Fatal Error - DOTNet with Core CAST AIP - ASP Net compilation failure could not find generated assembly
Failed to locate or open assembly	Either you don't have the complete code or the references are out of date.	Ensure that the code is complete and update the project references

Code Analysis

For code related issue we need to follow the same as above in the C#.NET

Memory Issues

For memory issue we need to follow the same as above in the C#.NET