CMS - Using the globallysharedsettings.ini file

Using the globallysharedsettings.ini file

This page describes how to add various parameters to the **GloballySharedSettings.ini** file (located in the following locations after installing CAST Products):

- **Windows XP/2003** > **%ALLUSERSPROFILE%\Application Data\CAST\CAST\<Version_Number>**
- **Windows Vista/7/2008** > **%ALLUSERSPROFILE%\CAST\CAST\<Version_Number>**

These parameters add supplementary features to the CAST Management Studio and especially useful if you are using the Command Line Interface (CLI):

- General
- ASP
- C++
- J2EE EJB
- J2EE
- .NET

General - Activating / deactivating the Inference Engine

Certain analyzers provide an Inference Engine to compute run time type information in order to simulate program behaviour during execution of the analyzer and thus identify additional links that would not otherwise be "discovered" using standard analysis techniques. This technology detects a reference to an object wherever its name is mentioned, regardless of the context in which this reference occurs.

The following analyzers incorporate an Inference Engine:

- **VB**
- **J2EE (for the Java files analysis)**
- **Mainframe (Cobol)**
- **T-SQL**
- **.NET Analyzer**

Activation

To activate the Inference Engine for all analyzers, use the following option (to deactivate it, use the "No" value, or remove the option entirely). By default this option is not present in the GloballySharedSettings.ini file.

```
[Inference Engine]Use Inference Engine=Yes
```

Trace Mode

To activate the trace mode, add the following parameter. By default this option is not present in the GloballySharedSettings.ini file. Use the "No" value to deactivate, if the option is present.

```
[Inference Engine]Use Inference Engine=YesACTIVATE_TRACE_INFENG=Yes
```

Limit Strings

To limit the number of strings that will be found during the search of each object value, add the following parameter. Note that limiting the number of strings can lead to incomplete results, however, performance of the Inference Engine is improved. By default this option is not present in the GloballySharedSettings.ini file.

```
[Inference Engine]Use Inference Engine=YesLimitString=5000
```

Note that when the Inference Engine is activated, but this option is not present in the GloballySharedSettings.ini file, the following default values are used:

- J2EE and T-SQL = **10000**
- VB **= 100000**
- Mainframe (Cobol) = **10**
- .NET = **3000**

Limit Sub Target

To limit the number of intermediate values that the Inference Engine must resolve in order to obtain the value of the object that is being searched for, add the following parameter. Note that limiting the number of intermediate values can lead to incomplete results, however, performance of the Inference Engine is improved. By default this option is not present in the GloballySharedSettings.ini file. The lowest value you can enter is 1.

```
[Inference Engine]Use Inference Engine=YesLimitSubTarget=200000
```

Note that when the Inference Engine is activated, but this option is not present in the GloballySharedSettings.ini file, the following default values are used:

- J2EE and T-SQL = **300000**
- VB **= 500000**
- Mainframe (Cobol) = **100**
- .NET = **3000**

Search for values through procedure calls

To deactivate the search of values through procedure calls, add the following parameter and "Yes" value. If the value is set to "Yes", then the search for values will be local to the procedure. By default this option is not present in the GloballySharedSettings.ini file. Adding the "No" value or removing the option will force the Inference Engine to search for values through procedure calls (the default setting):

```
[Inference Engine]Use Inference Engine=YesversionV1=yes
```

ASP

You can set various ASP Analyzer options via the **GloballySharedSettings.ini** file. For example:

- **Skip Analysis Saving**
- **Save Only File Structure**
- **Display Symbol Positions**
- **Display Saved Objects**
- **Display Saved Links**
- **Do Not Search For ActiveX**
- **Display Valid Searched URL**
- **Display Error Messages Only Once**
- **Display Analysis Time For Each File**
- **Display Analysis Time For Each Step**
- **Save Dynamic Parent Relationships**
- **Internet Connection Timeout = x** (use this option to optimize the Internet Connection Timeout for when ASP Analyzer tries to resolve external URL references in source code. By default, this is set to 4 seconds).

Add the options you want to activate or deactivate to the **GloballySharedSettings.ini** file as follows:

```
[WebAnalyzerOptions]Skip Analysis SavingDo Not Search For ActiveX = FALSEDisplay Valid Searched URL = 0Display
Error Messages Only Once
```

By default, adding = FALSE or = 0 will deactivate the option.

C++

```
[C++ Analyzer]OutputUnexpectedSyntax=yes
```

- **YES** > When unexpected code syntax is found by the analyzer during an analysis a warning message AND the unexpected syntax will be displayed in the Log Window.
- **NO** > When unexpected code syntax is found by the analyzer during an analysis a warning message will be displayed.

```
[C++ Analyzer]LastIncludePathSelected=YES
```

Saves the last Include Path selected in the wizard so that it can be re-used again. It is not necessary to modify this option.

```
[C++ Analyzer]LastRootPathSelected=YES
```

Saves the last Root Path selected in the wizard so that it can be re-used again. It is not necessary to modify this option.

```
[C++ Analyzer]KeepLastParsedFileTrace=filename
```

- where **<filename>** is equivalent to the full path (directory and file name) of the file containing the expanded (preprocessed) version of the last analyzed file. For example: **KeepLastParsedFileTrace=c:\Temp\CastPreprocessedFile.cpp**

This is usually the last file of the analysis, the file currently being analyzed or the file causing errors (where a fatal error occurs). This trace file can be particularly useful when there is some doubt about the way a macro has been expanded. By default, this option is not present in the **GloballySharedSettings.ini** file. As a result, you must specifically add the option to activate it, and then remove it to deactivate it.

Note that CAST also stores the file containing the expanded (preprocessed) version of the last analyzed file in:

%TEMP%\CAST\CAST\<Version_Number>\CAnalyzer2\PreprocessedFile.cpp

J2EE - EJB

Add the following option to the **GloballySharedSettings.ini** file if you want to retain proprietary XML files generated during the transformation process.

```
[EjbAssistant]KeepGeneratedFiles=yes
```

The transformation process is a necessary part of an EJB analysis. It converts files that contain beans persistence information from server specific format a format that can be analyzed correctly.

J2EE

File extraction from archive can be time consuming. By default, the J2EE Analyzer will extract in memory files that have been required from archives, in order to speed up the extraction process. This mode can be changed with the option **UnpackArchivesInMemory**:

```
[JavaAnalyzer]UnpackArchivesInMemory=yes
```

- Yes = extraction of archive files to memory
- No = extraction of archive files to disk

If **UnpackArchivesInMemory=no**, the analyzer will keep an extracted copy of files that have been required from archives. It then reuses them in consecutive jobs. Deleting these files has no effect on the analyzer. The next analysis will simply recreate them. However, they must not be modified in the meantime.

By default these extracted files are kept under the "Cast Generated Files" folder created under the default temp directory as specified by an environment variable. This location can be changed by setting the **DeployDir** option under the **JavaAnalyzer** section of the GloballySharedSettings.ini file to the desired directory. The specified directory must exist prior to analysis:

```
[JavaAnalyzer]DeployDir=<path>
```

The J2EE Analyzer can also perform the following checks during the analysis:

- In a Try and Catch statement, catch clause should not be empty ("empty catch clause")
- In a Try and Catch statement, Finally clause should not be empty ("empty finally clause")
- In a Try and Catch statement, the Finally clause should not contain a Return statement ("return in finally clause")
- Missing default in switch statement ("missing default in switch statement")

For example:

```
void f(){   try   {}   catch (Exception e)     <-- empty catch   {       ;   }   finally                 <--
empty finally   {       //TODO   }}
```

```
int g(){   int i;   switch (i)       <-- missing default   {       case 1: return 2;   }   try   {}   finally
{       return 1;    <-- return in finally   }   return 0;}
```

To activate the checks, add the following entry to the **GloballySharedSettings.ini**:

```
[JavaAnalyzer]ProcessAnomalies=1
```

.NET

If you have VB.NET or C# source code in ASP.NET 2.0 and abovce web applications that require compilation and you have selected the **"Legacy" .NET Analyzer**, an embedded web server is automatically started during the analysis process when analyzing ASP.NET 2.0 and above to compile the code. The web server is stopped when all ASP.NET pages that require compilation have been fully compiled.

By default, the web server runs on local port 2222, however, if you have existing services that conflict with this port number, you can override it by changing the port number via an option in the **.NET Analysis Unit editor** or by adding an entry in the **globallysharedsettings.ini** file. For example:

```
[DotNetWebServer]port_number=2223
```

If an existing service runs on local port 2222 an error message will be displayed in the log:

"AspNet compilation failure, could not start Cast web server on tcp port number 2222. If the port is already used by another process please consult the on-line help to change this port number."