|  |
| --- |
| Clone Detective for Visual Studio |
| Settings |
| Specification |

|  |
| --- |
| Immo Landwerth  7/26/2008 |

Table of Contents

[Goal 3](#_Toc205889247)

[System Settings 3](#_Toc205889248)

[User Interface 3](#_Toc205889249)

[Configuration Persistence 3](#_Toc205889250)

[Default Settings 3](#_Toc205889251)

[Solution Settings 4](#_Toc205889252)

[User Interface 4](#_Toc205889253)

[Configuration Persistence 4](#_Toc205889254)

[Default Settings 4](#_Toc205889255)

[Customizing Clone Detection 5](#_Toc205889256)

[Creating a custom clone detection analysis file 5](#_Toc205889257)

[Analysis File 5](#_Toc205889258)

[Clone result file 5](#_Toc205889259)

[ConQAT parameter overrides 5](#_Toc205889260)

[Macros 5](#_Toc205889261)

[Creating a Custom ConQAT Analysis File 6](#_Toc205889262)

[Appendix 7](#_Toc205889263)

[Determining Java Home 7](#_Toc205889264)

# Goal

The goal of this document is to specify how Clone Detective should support customizations in order to better fit the needs for both existing ConQAT users as well as new users that just quickly want to run clone detection.

# System Settings

## User Interface

The user can customize the system settings by using the Tools -> Options -> Clone Detective -> System Settings panel of the Settings dialog. Figure 1 shows the contents of this panel.



Figure

## Configuration Persistence

The system settings are stored in the registry under this hive:

HKCU\Software\Microsoft\VisualStudio\9.0\DialogPage\CloneDetective.Package.CloneDetectiveOptionPage

The names of string values are

ConqatFileName

JavaHome

## Default Settings

If the user has not manually configured the system settings (or has explicitly set both values to an empty string) the following default values are used:

* **ConQAT batch file**. If ConQAT has been installed together with Clone Detective the path is set to the installation directory of ConQAT suffixed with bin\ConQAT.bat. If ConQAT has not been installed together with Clone Detective an error message will be displayed stating that the user must manually enter the correct path.
* **Java home**. The Java home is determined by looking at the registry (see for details). If the path could not be determined an error message will be displayed stating that the user must manually enter the correct path.

# Solution Settings

## User Interface

The user can customize the solution settings by using the Tools -> Options -> Clone Detective -> Solution Settings panel of the Settings dialog. Figure 2 shows the contents of this panel.



Figure

## Configuration Persistence

The clone detective configuration is stored per user and solution. That means the configuration is stored in a file named like this

$(SolutionName).CloneDetective.user

For example, for a solution named MySolution the settings file would be named MySolution.CloneDetective.user.

Since Visual Studio uses a default exclude of \*.user for files put under version control that settings file does not accidentally get checked-in and therefore remains per user.

## Default Settings

If a solution is opened the first time no custom clone detection settings exist. In this case Clone Detective will use default settings. The settings page represents this by the check box named *Use custom clone detection*. If this check box is not marked Clone Detective uses the default settings as displayed in Figure 2.

A default ConQAT analysis file is provided with the installation of Clone Detective and resides in the installation directory (e.g. C:\Program Files\Clone Detective\DefaultCloneDetection.cqa). This ConQAT analysis file provides four parameters:

|  |  |
| --- | --- |
| Parameter | Description |
| solution.dir | The fully qualified path to the solution directory. All source code files contained in this directory (or any subdirectory) are analyzed for clones. |
| output.dir | The fully qualified path to the directory the clone results file is written to. |
| output.file | The name of the clone results file that is written by the clone detection. |
| clone.minlength | The minimum lines of code a clone must have to be considered. |

## Customizing Clone Detection

You can customize clone detection by checking the check box *Use custom clone detection*. In this case, all controls in Clone Detective Settings dialog are enabled, but their current values are preserved. This allows a step-by-step customization starting from the default configuration.

Figure 3 shows the configuration that is created by just checking *Use custom clone detection*. Figure 4 shows a completely customized example.

|  |  |
| --- | --- |
| Figure | Figure |

### Creating a custom clone detection analysis file

The *Create* button will start a wizard that will create a custom clone detection analysis file (see *Creating a Custom ConQAT Analysis File* for details).

### Analysis File

In this text box you must specify the fully qualified of the ConQAT analysis file. Figure 3 shows a configuration that uses the default file while Figure 4 shows a configuration that uses a custom file.

### Clone result file

In this text box you must specify the fully qualified path of the clone results XML file that will be produced by the ConQAT analysis.

### ConQAT parameter overrides

This list view shows all properties declared in the ConQAT analysis file. You can override them by clicking the check box and providing a value. These values are passed to ConQAT using the -p command line switch.

### Macros

You can only use fully qualified (i.e. absolute) paths for both the ConQAT analysis file and the clone result file. If you want to specify paths that are relative to the solution you must use one of the macros listed in the table below.

The macros are also available in the property overrides section.

|  |  |
| --- | --- |
| Macro | Example Value |
| $(InstallDir) | C:\Program Files (x86)\Clone Detective\ |
| $(ConQATDir) | C:\Program Files (x86)\Clone Detective\ConQAT\ |
| $(DevEnvDir) | C:\Program Files (x86)\Microsoft Visual Studio 9.0\Common7\IDE\ |
| $(SolutionPath) | D:\MySolution\Solution1.sln |
| $(SolutionDir) | D:\MySolution\ |
| $(SolutionFileName) | Solution1.sln |
| $(SolutionName) | Solution1 |
| $(SolutionExt) | .sln |
| $(ConfigurationName) | Debug |
| $(PlatformName) | Any CPU |

# Creating a Custom ConQAT Analysis File

Currently, to create a custom clone detection file you have to copy and paste the contents of DefaultCloneDetection.cqa into a new file that must be named CloneDetection.cqa and has to reside in the same folder as the solution. Now, as described in *Creating a custom clone detection analysis* file, Clone Detective will provide a *Create* button that starts a wizard to perform this action. This section shows the pages of the wizard.

|  |  |
| --- | --- |
| Figure | Figure |
|  |  |
| Figure | Figure |

# Appendix

## Determining Java Home

To determine the current Java home Clone Detective first tries to open the following registry key:

HKLM\SOFTWARE\JavaSoft\Java Runtime Environment

Then it tries to read the string value named

CurrentVersion

With the resulting value (named <Version>) it tries to open the following registry key:

HKLM\SOFTWARE\JavaSoft\Java Runtime Environment\<Version>

Then it tries to retrieve the value of the string key

JavaHome