

Quick Reference

Template File Summary

<u>Template File</u>	<u>Title</u>
CPPModsRec.ecgt	General CPP Modifications Record
CPPSourceFiles.ecgt	General CPP Class Source Files
DataFileReg.ecgt	Data File Registration Entry File
ECGTSourceFile.ecgt	ETAC Code Generator Template Source File
ETACForApp.ecgt	ETAC Interpreter Access for an Application Program
ETACMainApp.ecgt	Main ETAC Application Program Source File
ETACModsRec.ecgt	General ETAC Modifications Record
ExternTACLib.ecgt	External TAC Library Source File
MakeEXE.ecgt	EXE File Creator
MakeExtractor.ecgt	ZIP File Self-extractor

ETAC Code Generator Templates

1 August 2020

Copyright © **Victor Vella** (2019-2020)
All rights reserved.

Contents

Quick Reference

Contents

Document Conventions

- 1. Introduction**
- 2. Template Files**
- 3. Editing Template Files**
- 4. The Templates Folder**
 - 4.1 CPP Modifications Record
 - 4.2 CPP Source Files
 - 4.3 Data File Registration
 - 4.4 ECGT Source File
 - 4.5 ETAC for Application Programs
 - 4.6 Main ETAC Application Program
 - 4.7 ETAC Modifications Record
 - 4.8 External TAC Library
 - 4.9 Make Executable
 - 4.10 Make File Extractor
- 5. The ECGData Folder**

Bibliography

Glossary

Legal Information

 (the ETAC Code Generator logo) is an unregistered trademark (™) of Victor Vella.

ETAC is an unregistered trademark (™) of Victor Vella for *computer software incorporating an implementation of a computer programming language*. There may be other owners of the “ETAC” trademark used for other purposes.

Unicode is a registered trademark (®) of Unicode, Inc. in the United States and other countries.

This document is copyright © by Victor Vella (2020). All rights reserved. Permission is hereby granted to make any number of exact electronic copies of this document without any remuneration whatsoever. Permission is also granted to make annotated electronic copies of this document for personal use only. Except for the permissions granted, and apart from any fair dealing as permitted under the relevant Copyright Act, no part of this document may be reproduced or transmitted in any form or by any means without the express permission of the author. The copyright of this document shall remain entirely with the original copyright holder.

The author of this document shall not be liable for any direct or indirect consequences arising with respect to the use of all or any part of the information in this document, even if such information is inaccurate or in error. The information in this document is subject to change without notice.

Document Conventions

The following symbolic conventions are used in this document.

<u>Symbol</u>	<u>Meaning</u>
$\langle x \rangle$	separates x as a unit of information from the surrounding text.
...	represents omitted text (as usual).
<i>text</i>	maroon coloured italic text is a link to the text's definition.
<u>text</u>	underlined green text is a link into the document.

ETAC Code Generator Templates

This document is for version **2-0-3-ena** of the **ETAC Code Generator** program (ETACCodeGen.btac), which is compatible with the **ETAC Programming Language** version **1-1** (implemented in program RunETAC.exe version **3-0-6-ena**).

(Australian English)

1. Introduction

This document describes the *template files* that are released with the **ETAC Code Generator**. The files exist under a folder called `Templates` in the installation folder. *Template files* have an extension of `‘ecgt’`, and can be used with ETACCodeGen.btac and ETACCodeGen.exe. For information on how to use the **ETAC Code Generator** see the document *ETAC Code Generator* (ETACCodeGenerator.pdf).

Changes from Previous Publication

The following section indicates the changes made in this publication from the previous publication (5 January 2019). Most of the changes are adaptations to Unicode®.

Modified Items

[ECGTSourceFile.ecgt](#) - [ETACMainApp.ecgt](#) - [ExternTACLib.ecgt](#)

2. Template Files

Each *template file* requires *template arguments*, which are described in the *template file* itself. Some *template files* have the optional keywords `<*INS>` and `<*OUT>` as part of their *template arguments*.

The `<*INS>` keyword is intended to be used for inserting code that is generated by another *template file* to be inserted into generated text. For example, the `<*INS>` *template argument* in the CPPModsRec.ecgt *template file* allows the code generated by that *template file* to be inserted into files generated by other *template files* such as the CPPSourceFiles.ecgt *template file*. The CPPSourceFiles.ecgt *template file* contains the text line shown below, which generates the text from CPPModsRec.ecgt into the text generated from CPPSourceFiles.ecgt.

```
<@GEN: [ INPUT="CPPModsRec" INSERT ARGS=(FILE=<CLASS_NAME>.h *INS) ] >
```

The **ETAC Code Generator** automatically replaces the line above by the C++ modifications record generated from CPPModsRec.ecgt, but without the extra descriptive information.

The `<*OUT>` keyword is intended to be used to generate portions of text existing in a *template file* that are to be output to their appropriate files, rather than those portions being generated into the *output file*. For example, the `<*OUT>` *template argument* in the CPPSourceFiles.ecgt *template file* allows the generated text to be output into the user-specified C++ header and source files. If the user does not specify the `<*OUT>` keyword when using CPPSourceFiles.ecgt, then the generated text will be put into the single *output file* as follows.

```
///// Start of file ...h
...
///// End of file ...h

///// Start of file ...cpp
...
///// End of file ...cpp
```

Sometimes it is convenient for the user to allow the generated text to be put into the single *output file* rather than be put into their appropriate text files.

The *template files* in the Templates folder and data files in the ECGData sub-folder, and their *generated files*, can be modified by the user subject to the licence agreement.

Note that ECGFile.ico (which exists in the Res folder of the **ETAC Code Generator** installation) can be used as an icon for *template files*; the user will need to set up such usage himself.

3. Editing Template Files

Any text editor can be used to edit *template files*. However, an editor that has syntax highlighting and folding would be of great advantage. One such editor is the free **Notepad++** editor (obtainable from the World Wide Web), which can be used to edit other programming language files as well. The **ETAC Code Generator** installation includes one file, ECGTdef.xml, that can be used with **Notepad++**. The file exists in the <...\ETACCodeGenerator\Other> (or <...\ETACCodeGenEXE\Other>) folder.

ECGTdef.xml is the **Notepad++** definition file for syntax highlighting the contents of *template files* displayed in the **Notepad++** editor window. Within **Notepad++**, select the menu 'Language→Define your language...'. Then 'Import...' the ECGTdef.xml file.

Note that the **Notepad++** editor is not completely suited for displaying *ECGL* text properly when using the ECGTdef.xml file; the editor may show some incorrectly highlighted items.

4. The Templates Folder

The Templates folder can be relocated to a different location within the file system. If the folder is relocated, the contents of the file ETACCodeGen.ini may need to be updated to indicate the new location. MakeECGIni.cmd can be used to update ETACCodeGen.ini if the installation folder itself is renamed or moved to a different location.

The following is a summary of the contents of the Templates folder within the **ETAC Code Generator** installation folder.

4.1 CPP Modifications Record

Template: [CPPModsRec.ecgt](#)

This template generates a modifications record for source files written in the C++ programming language. The template is typically used with *template files* that generate C++ source files to produce the modifications record for those source files. A typical usage inserted into a *template file* is as follows.

```
<@GEN: [ INPUT="CPPModsRec" INSERT ARGS=(FILE=... *INS) ]>
```

The line above generates and inserts a modifications record into (the internal copy of) the *template file* containing the line.

4.2 CPP Source Files

Template: [CPPSourceFiles.ecgt](#)

This template generates the skeleton code for a general C++ source file and header file for a class structure.

4.3 Data File Registration

Template: [DataFileReg.ecgt](#)

This template generates a registry text file (*.reg) to register and implement features for a file type. The following optional features are implemented.

- Default icon for the specified file type.
- New file creation via the context menu 'New' entry.

- Open with list via the context menu ‘Open With’ entry.
- Action verbs via the context menu.

Note that only the registry file is created, the system registry itself is unaltered.

4.4 ECGT Source File

Template: [ECGTSourceFile.ecgt](#)

This template generates a skeleton *template file* for use with the **ETAC Code Generator**.

4.5 ETAC for Application Programs

Template: [ETACForApp.ecgt](#)

This template generates a C++ skeleton source file to interact with the ETAC interpreter via AppETAC.dll (supplied with the **Run ETAC Scripts** package).

4.6 Main ETAC Application Program

Template: [ETACMainApp.ecgt](#)

This template generates a skeleton ETAC text script file (*.etac) for creating an ETAC application program that has a dialog box as the main window.

4.7 ETAC Modifications Record

Template: [ETACModsRec.ecgt](#)

This template generates a modifications record for source files written in the ETAC programming language. The template is typically used with *template files* that generate ETAC source files to produce the modifications record for those source files. A typical usage inserted into a *template file* is as follows.

```
<@GEN: [ INPUT="ETACModsRec" INSERT ARGS=(FILE=... *INS) ] >
```

The line above generates and inserts a modifications record into (the internal copy of) the *template file* containing the line.

4.8 External TAC Library

Template: [ExternTACLib.ecgt](#)

This template generates or updates a C++ source file for compiling into an external TAC library (ETL) for use with ETAC programs.

4.9 Make Executable

Template: [MakeEXE.ecgt](#)

This template generates a NSIS source file (*.nsi) for producing an executable file (*.exe) from a computer program and associated files contained in a directory. A pre-existing NSIS compiler (obtainable free from the World Wide Web) is required to produce the final executable file.

The template can be used to convert programs written in the ETAC programming language into executable files (*.exe) to run on computers that do not have the **Run ETAC Scripts** package installed.

4.10 Make File Extractor

Template: [MakeExtractor.ecgt](#)

This template generates a NSIS source file (*.nsi) for extracting files contained in a directory. A pre-existing NSIS compiler (obtainable free from the World Wide Web) is required to produce the final self-extracting executable file.

5. The ECGData Folder

This folder exists under the `Template` folder, and contains data files that are used by the *template files*. The two files in this folder, `ExternTACLibECG.cpp` and `ExternTACLibECG.def`, are used by the *template file* `ExternTACLib.ecgt` as initial source files.

Bibliography

ETAC Code Generator copyright © Victor Vella (2020).

The Official ETAC Programming Language copyright © Victor Vella (2020).

Glossary

E

ECGL

ECGL stands for “**ETAC Code Generator Language**”. *ECGL* is a unique sophisticated declarative *template file* language used by the **ETAC Code Generator**.

G

generated file

A disk file generated or modified by the **ETAC Code Generator**.

O

output file

The file that receives the default generated text of a *template file*. This is the main file produced by the **ETAC Code Generator**. The data in the *output file* is, in fact, a modified copy of the *template file* after all processing has been completed.

T

template arguments

Template arguments are keywords and their arguments, supplied by the user, that match the template specification (as defined by the keyword-arguments system) of a *template file*.

See **A.3 Source String Syntax** under **Appendix A: Keyword-arguments Specification** in the document “*The Official ETAC Programming Language*” (ETACProgLang(Official).pdf) for more details about the syntax of *template arguments*.

template file

A special text file, for use with the **ETAC Code Generator**, acting as a model for generating or modifying one or more text files for desired purposes. A *template file* contains fragments of the generated text, special codes, and possibly ETAC script, that describe how those fragments are to be produced into the *generated files*.