

Think Class Library 2.0 CodeWarrior Port Package

Version 1.6 • 30 November 1994
Produced by Eric Scouten

Note

This document was produced using Microsoft Word and the TrueType fonts Palatino, Helvetica, Courier, and Zapf Dingbats. There are two versions of this document: `**README.word` which is a Microsoft Word 5 document, and `**README.standalone` which is a combined application/document created with Print2Pict. ❖

Contents

Chapter 1 About the TCL Port Package 1

- Introduction 1
- Package Requirements 1
- Overview 2
- For More Information 2

Chapter 2 Converting the TCL Source Code 3

- Creating the Modified TCL Source Code 3
- Previous Users of the TCL Port Package 4

Chapter 3 Converting Your Application Source Code 5

- Creating a Project File 5
 - Building a New Project File 5
 - Project Preferences 6
- Changes in Coding Technique 6
 - C++ Templates 6
 - Exception Handling 6

Chapter 4 Miscellaneous 8

- Version History 8
- Acknowledgments 9

About the TCL Port Package

Introduction

This document describes the TCL 2.0 CodeWarrior Port Package, a set of files which will enable you to compile the Think Class Library version 2.0 with the Metrowerks CodeWarrior 68K and PowerPC compilers. Read this document carefully **before** using the package.

This package will be updated regularly as new versions of the Metrowerks compilers become available and as new bug fixes or features are incorporated. The most recent version should always be available for FTP on the site `daemon.ncsa.uiuc.edu` in the directory `/pub/TCL/contributors/Eric_Scouten`. Notification of these updates will be posted to the Internet newsgroup `comp.sys.mac.oop.tcl`.

This package **does not include** the Think Class Library itself, in either original or modified form. You must purchase a copy of Symantec C++ for Macintosh, version 7.0, in order to use the TCL.

Note

Additional information may be contained in a file named `*README.extra` in the root folder of the package. ❖

Package Requirements

To make effective use of the CodeWarrior Port Package, you need the following items:

- Symantec C++ for Macintosh, version 7.0.
- Metrowerks CodeWarrior Bronze or Gold, CW/5 or newer.
- At least 30MB disk space available on your hard drive.
- A 680x0 Macintosh with at least 8MB of RAM, or a PowerPC Macintosh with at least 16MB of RAM.
- An unmodified copy of the Think Class Library, version 2.0.3. You will need the free updater provided by Symantec to upgrade the TCL from version 2.0 to version 2.0.3. This is available for anonymous FTP from the site **devtools.symantec.com** as part of the Symantec C++ updater to version 7.0.4. (It may also be available on other FTP sites.) Note: No update to the TCL was included with the 7.0.4 patch set; it is the same as the 7.0.3 version of the TCL.

The following items **will not** be useful:

- Think C or Symantec C++, version 6.0 or earlier.
- Think Class Library, version 1.1.3 or earlier. (If you are using TCL 1.1.3, you may wish to investigate the PTR-TCL package provided by Jon Wätte on common FTP sites.)
- The free updater provided by Symantec to upgrade Symantec C++ from version 6.0 to version 7.0. The updater does not include the TCL version 2.0.

Overview

Using the TCL CodeWarrior Port Package entails the following steps:

- Converting the Think Class Library to a form usable by the Metrowerks compilers.
- Converting your source code to a form usable by the Metrowerks compilers.

The following chapters of this document describe these steps in detail. You should also review the chapter entitled, “Using the Modified TCL,” which describes the differences between the standard TCL used for the Symantec compiler and the modified TCL used for the Metrowerks compilers.

For More Information

If you have questions about using the TCL CodeWarrior Port Package, you may contact me at the following addresses:

Internet	scouten@uiuc.edu
Telephone	+1 217 333 5214 (office) +1 217 344 1275 (home)
Fax	+1 815 836 3221
US mail	Eric Scouten 404 N Lake St Urbana, IL 61801

Please note that this package is provided **free of charge**. Although I am generally available (and interested) to talk about this package and any related topic, I may from time to time be unable to respond to requests for support of the package due to other commitments.

Creating the Modified TCL Source Code

In this stage, you will create a second copy of the Think Class Library source code on your hard drive. You should keep two copies of the TCL source code on your drive: the original source for use when compiling with Symantec's compiler, and the modified source for use with CodeWarrior. (The TCL source and precompiled headers take approximately 6MB.)

❖ WARNING

It is absolutely important that you begin with the **unmodified** source code from the TCL 2.0.3. If you have modified the TCL yourself, make a copy of your changes and reinstall the TCL from your Symantec C++ 7.0 master disks. Then update the TCL source with the updater files provided by Symantec to version 2.0.3. (If later versions of the TCL are released, **do not** use these versions either.) ❖

Converting the updated TCL requires the following steps:

- Make a copy of the Think Class Library 2.0 folder. Place this folder **outside** the CodeWarrior compiler folder (and outside the Symantec C++ folder as well.)
- Open the MultiDiff application. Select "Apply Diff..." from the File menu. A standard open box will appear. Select the file ***TCL source.diff**. You will be prompted for the top folder to apply the patches to. Select the folder which contains the copy of the TCL you just made. Note that the changes are made **in place**; thus it is important to make a copy of the TCL prior to applying the patches.
- Wait. There is no way to stop the patcher once it's running. It will take several minutes to make the changes. (On a PowerMac 6100/60 it takes about 4 minutes.)
- Copy the header files **THINK.h** from your Symantec C++ folder to your CodeWarrior folder. If you maintain separate folders for 68K and PPC compilers, copy the header to both folders. (**THINK.h** can be found in the path **:Symantec C++ for Macintosh:Mac #includes:THINK #includes**.)
- Copy the header files **SANE.h** from your Symantec C++ folder to your CodeWarrior folder. If you maintain separate folders for 68K and PPC compilers, copy the header to the 68K compiler folder only. **Rename** the header to **ThinkSANE.h**. (SANE.h can be found in the path **:Symantec C++ for Macintosh:Mac #includes:THINK #includes**.)

If you are making use of the Visual Architect, you must perform the following additional steps:

- Make a copy of the Visual Architect Macro Files folder. Place this folder **outside** the CodeWarrior compiler folder (and outside the Symantec C++ folder as well.) This will be your backup copy of the macro files.
- Open the MultiDiff application. Select “Apply Diff...” from the File menu. A standard open box will appear. Select the file ***VA macros.diff**. You will be prompted for the top folder to apply the patches to. Select the **original copy** of the Visual Architect Macro Files folder. (This is necessary so that future projects built with VA will pick up the changes.) Note that the changes are made **in place**; thus it is important to make a copy of the TCL prior to applying the patches.
- Wait. This patch should only take a minute or two.

Previous Users of the TCL Port Package

If you have been using previous versions of the TCL port package, there are a few changes that you will need to be aware of. This version of the TCL port package is designed to be used **only** with CodeWarrior CW/5 or newer and takes advantage of several new features in this version of the compiler. You will need to follow these steps to adapt your code for this version of the port package:

- Build a new version of the modified TCL as described in the previous steps.
- Add the file **CW TCLHeaders.pch** to your CodeWarrior project (both 68K and PowerPC projects, if applicable). This will cause the precompiled headers to be automatically generated whenever necessary.
- Check for newly undefined symbols. The CW/5 release includes a new version of Apple’s Universal Headers. Several symbols were renamed in this version of the headers. (This note is not specific to the TCL port package.)
- If you use templates (especially if you used the Template Munger), you may need to make a few minor adjustments to your code. Support for the macro substitution for templates has been removed. Your existing code should compile essentially “as is:” the old `TCL_NO_TEMPLATES` macro is now undefined (and thus false). You may need to add explicit class declarations (`class CList<CMyClass>;`) to force CodeWarrior to generate certain template instantiations; this is similar to Symantec’s `#pragma template CList<CMyClass>` statement.

Creating a Project File

You will find it easiest to start from the project stationery files which were included with the port package. These projects loosely follow the segmentation in the sample TCL project files provided by Symantec.

Building a New Project File

If you choose to create your own project file, you will need to be aware of a few considerations. The virtual function tables generated by the TCL tend to be rather large. For small TCL projects, you will be able to compile “as is.” If you have more than a few of your own classes, you will need to turn on “Far virtual function tables” in the Preferences Language panel.

To create the list of source files, you may simply include all of the files in the modified TCL source directory. Please note the following differences between the CodeWarrior projects and Symantec C++ projects:

Removed Source Files

CSaver.cpp
CStack.cp
PutObject1.cpp
BRLib.n

Added Source Files

BRClasInf.cpp
BRPriStr.cpp
UDebugging.cp (from Metrowerks PowerPlant – Utility folder)
UExceptions.cp (from Metrowerks PowerPlant)

Project Preferences

You will need to set a few special project preferences for working with the modified TCL. These are:

- **Prefix file** (Language panel): CW TCLHeaders68K or CW TCLHeadersPPC
- **Access paths** (Access paths panel): Add folder where modified TCL is housed
- **Treat #include <...> as #include "..."** (Access paths panel): Must be checked if using Visual Architect files

Changes in Coding Technique

This section describes some differences between the Symantec C++ version of the TCL and the modified TCL for CodeWarrior.

C++ Templates

The Metrowerks implementation of templates has some subtle differences from Symantec's implementation. Symantec C++ relies on two pragmas to control the instantiation of templates: `#pragma template_access` and `#pragma template`. These pragmas are not recognized (and not necessary) with CodeWarrior.

For the most part, CodeWarrior will automatically generate the template instantiations that you need for your code if you make the template definition (.tem file) available where the template is actually used. In those cases where it does not (or where you wish to have the templates compiled separately), you may force a template instantiation by adding an explicit reference (`class my_template<my_class>;`).

If you have been using prior versions of the TCL port package, you can now remove the ".tm.h" files which were generated by Template Munger and the associated macros in your source files. The `TCL_NO_TEMPLATES` macro is no longer defined.

Exception Handling

The Bedrock Exception Library which was added to the TCL in version 2.0 seems to be somewhat buggy, especially on the PowerPC. For this reason, I have chosen to replace the BEL with the exception library which comes with PowerPlant. This change is triggered by a macro defined in the **CW HeadersTCL.h** file:

```
#define TCL_USE_PP_EXCEPTIONS
```

If you wish to retain the BEL in place of the PowerPlant library, you may comment out this macro.

You may continue to use all of the macros defined for the TCL without changing your source code, with the following restrictions:

- You may not use the typed exception handling which was added in the TCL 2.0. The macros `catch_`, `catch_reference_`, and `catch_no_instance_` are not permitted. You must use `catch_all_` instead.
- You may not reference the global variables `gAskFailure`, `gBreakFailure`, `_gTCLBreakCatch`, `_gTCLBreakFailure`, or `_gTCLBreakAssert`. These variables do not exist in this implementation. The macros which access these variables are implemented.
- The PowerPlant header `<UException.h>` must be installed in the CodeWarrior compiler folder. The file `<UException.cp>` must be added to the project.
- If the `__TCL_DEBUG__` macro is defined, the PowerPlant header `<UDebugging.h>` must also be installed in the CodeWarrior compiler folder. The file `<UDebugging.cp>` must be added to the project.
- Your source may not use the `throw_(exception)` macro. Error reporting must take place through the standard `Fail___` routines. The `throw_same_()` macro is permitted.

Note

The run-time type identification (RTTI) mechanism of the BEL is still used in the port package. ❖

Version History

Version 1.0, released 9 May 1994:

- Port package for CodeWarrior DR/2.

Version 1.2, released 26 May 1994:

- Updated for CodeWarrior DR/3.
- Replaced Bedrock exception library with PowerPlant exception library.
- Improved documentation.
- Skipped version 1.1 to avoid confusion with the TCL of the same name.

Version 1.2.1, released 27 May 1994:

- Quick fix to include a missing file (which wasn't really missing, anyway).

Version 1.3, released 30 May 1994:

- Updated for TCL version 2.0.2.
- Included Template Munger application.

Version 1.4, released 21 June 1994:

- Checked for compatibility with CodeWarrior DR/3.5.
- Included template macro files with the patch file (as was the case in version 1.2 of the port package).
- Corrected bugs with object I/O and CStream.
- Tested with Visual Architect classes and functions. Added diffs for the VA macros.

Version 1.5, released 29 June 1994:

- Updated for TCL version 2.0.3.
- Fixed problems with diffs files in 1.4.

Version 1.5.1, released 17 July 1994:

- Updated for compatibility with CodeWarrior CW/4. (Minimal changes.)
- Fixed casting bug in CApplication.cp.
- Improved drawing of popup menus on PowerPC builds.

Version 1.5.2, released 10 September 1994:

- In project files included with version 1.5.1, some segments were incorrectly marked as purgeable. Fixed.
- Updated contact address.

Version 1.5.3, released 11 October 1994:

- Updated for compatibility with CodeWarrior CW/4.5. Fixed a problem with source file alloc.cp generating “illegal function overloading” errors. (Don’t ask how!)
- Fixed problem with CSaver template reference to GetObject.

Version 1.6 released 30 November 1994:

- Updated for compatibility with CodeWarrior CW/5.
- Uses templates instead of macro substitutions. Template Munger application removed.
- Uses automatic precompiled headers (.pch).
- Uses the new Universal Headers, version 2.0.

Acknowledgments

A special thanks to Jon Wätte who provided the MultiDiff application and inspired (challenged?) me to attempt this port.

I am especially grateful to the several people at Metrowerks who have supported this effort: Jonathon Hess, Lisa Lee, Andreas Hommel, Greg Galanos, Berardino Baratta, and Mark Anderson.