A Work-In-Progress report
by Paul Obermeier

BAWT

= 

+ 

European Tcl/Tk User Meeting 2016, Eindhoven
• BAWT Introduction
  • Intention
  • Requirements
  • Workflow Overview

• BAWT Examples
  • Build Tcl BI and Tclkits
  • Build libraries for Tcl3D

• BAWT Status
  • Overview of supported procedures
  • Build on Windows and Unix
  • What’s on the TODO list?

• BAWT Summary
Heterogeneous environments on Windows for building 3\textsuperscript{rd} party libraries and tools

- configure / make via MSYS / MinGW
- nmake
- cmake
- Visual Studio Solutions
- gcc via MSYS / MinGW

I want to automatically build all needed C/C++ libraries without user interaction.
### BAWT Introduction – Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Why and How</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support multiple Visual Studio versions.</td>
<td>Check and test before upgrading to a new version.</td>
</tr>
<tr>
<td>Support DOS and MSys build environment from 1 command shell.</td>
<td>Necessary for automated builds (ex. nightly bulds).</td>
</tr>
<tr>
<td>Make build and distribution process flexible.</td>
<td>Adapt to individual needs (private, work) using Tcl. Generate ready-to-use directories both for a developer and a user.</td>
</tr>
<tr>
<td>Keep library source code in compressed files.</td>
<td>Easy and fast download from webspace or repository</td>
</tr>
</tbody>
</table>
BAWT Introduction – Workflow

**Source directory**
- Library source codes
- Library build scripts

**BAWT Framework**
- BAWT build script
- Configuration scripts

**Build directory**
- Development folder
- Distribution folder

---

**BAWT – Build Automation With Tcl**
Paul Obermeier, paul@poSoft.de
BAWT Examples – Build Tcl BI and Tclkits

**Source directory**

- Bawl-All
- Bawl-Tcl
- Bootstrap-Darwin
- Bootstrap-Windows
- InputLibs
- Build.bawl
- Build-Darwin.sh
- Build-Linux.sh
- Build-Windows.bat
- tclkit-Darwin64
- tclkit-Linux32
- tclkit-win32.exe

**BAWT Stages**


**Build directory**

- BAWT - Bootstrap tools for Windows
  - 7-Zip.zip
  - gcc4.9.2_i686-w64-mingw32.7z
  - gcc4.9.2_x86_64-w64-mingw32.7z

- BAWT - Bootstrap tools for Darwin
  - 7-Zip.zip

- BAWT - Source packages and setup scripts

<table>
<thead>
<tr>
<th>Source</th>
<th>Setup</th>
<th>Build type on Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tcl/Tk Core</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tcl 8.6.5.7</td>
<td>Tcl bawl</td>
<td>MSys / gcc</td>
</tr>
<tr>
<td>Tk 8.6.5.7</td>
<td>Tk bawl</td>
<td>MSys / gcc</td>
</tr>
<tr>
<td>Tcl8.6.5.7</td>
<td>Tcl8.6.5.7</td>
<td>MSys / gcc</td>
</tr>
<tr>
<td>Tk8.6.5.7</td>
<td>Tk8.6.5.7</td>
<td>MSys / gcc</td>
</tr>
<tr>
<td>Tcl8.6.5.7</td>
<td>Tcl8.6.5.7</td>
<td>Compiled Tcl packages</td>
</tr>
<tr>
<td>Tckit</td>
<td>Tckit bawl</td>
<td>MSys / gcc</td>
</tr>
</tbody>
</table>

*Manual work: Download BAWT framework and appr. bootstrap files from www.bawl.tcl3d.org*
Example: Build Tcl BI and Tclkits for 32-bit Windows

- cd \temp\Bawt-0.10\Bawt
- Build-Windows.bat x86 Windows Bawt-Tcl\TclRuntime.tcl

```
CreateImportLibs false

# Setup libName         zipFile           scriptFile       buildTypes
# Setup Tcl and Tk.
Setup Tcl        Tcl-8.6.5.7z         Tcl.bawt         Release
Setup Tk         Tk-8.6.5.7z          Tk.bawt          Release

# Compiled Tcl packages.
Setup TkImg      TkImg-1.4.6.7z       TkImg.bawt       Release
Setup tcludp     tcludp-1.0.9.7z      tcludp.bawt      Release
Setup tDOM       tDOM-0.8.3.7z       tDOM.bawt        Release
Setup Tktable    Tktable-2.10.7z     Tktable.bawt     Release

# Pure Tcl packages.
Setup tablelist  tablelist-5.15.7z   tablelist.bawt    Release

# Tclkits.
Setup Tclkit     Tclkkit-8.6.5.7z     Tclkkit.bawt     Release
```

Configuration file: TclRuntime.tcl
BAWT Examples – Build Tcl BI and Tclkits

Source directory

Build directory

BAWT Stages

Bootstrap

Setup

Clean

Extract

Configure

Compile

Distribute

Action: Extract and copy bootstrap tools into build directory

Bootstrap

ZIP file: C:/temp/Bawt-0.1.0/Bawt/Bootstrap-Windows/7-Zip.zip

ZIP file: C:/temp/Bawt-0.1.0/Bawt/Bootstrap-Windows/gcc4.9.2_i686-w64-mingw32.7z

Adjust fstab files for MSys/MinGW gcc4.9.2_i686-w64-mingw32

Skipping C:/temp/Bawt-0.1.0/Bawt/Bootstrap-Windows/gcc4.9.2_x86_64-w64-mingw32.7z
BAWT Examples – Build Tcl BI and Tclkits

Source directory

BAWT Stages

Build directory

Setup tcludp
ZIP file : C:/temp/Bawt-0.1.0/Bawt/InputLibs/tcludp-1.0.9.7z
Version : 1.0.9
Build types: Release

Action: Check existence of source and setup files and download, if not existent.
Clean tcludp (Release)
DirDelete
Directory: C:/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Build/tcludp
DirDelete
Directory: C:/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Install/tcludp

**Action:** Remove library specific build and install directory.
BAWT Examples – Build Tcl BI and Tclkits

Source directory

BAWT Stages

Build directory

ExtractLibrary
ZIP file: C:/temp/Bawt-0.1.0/Bawt/InputLibs/tcludp-1.0.9.7z
Target directory: C:/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Build/tcludp

Action: Extract library source code into build directory.
BAWT Examples – Build Tcl BI and Tclkits

**Source directory**

**BAWT Stages**

- Bootstrap
- Setup
- Clean
- Extract
- Configure
- Compile

**Build directory**

```
MSysRun
Shell: C:/temp/Bawt-0.1.0/BawtBuild/Tools/gcc4.9.2_1686-w64-mingw32/msys/bin/sh.exe
> cd /C/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Build/tcludp
> /C/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Build/tcludp/configure
   --enable-shared
   --build=1686-w64-mingw32
   --prefix=/C/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Install/tcludp
   --exec-prefix=/C/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Install/tcludp
   --with-tcl=/C/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Build/Tcl
   --disable-symbols
Status: OK
```

**Action:** Configure library for compilation.
BAWT Examples – Build Tcl BI and Tclkits

Source directory

BAWT Stages

Build directory

MSysBuild
Build directory : /C/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Build/tcludp
Build target : install-binaries

MSysRun
Shell: C:/temp/Bawt-0.1.0/BawtBuild/Tools/gcc4.9.2_i686-w64-mingw32/msys/bin/sh.exe
> cd /C/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Build/tcludp
> make -j 4
> make install-binaries
Status: OK (Messages have been written to stderr)

Action: Compile and install library.
BAWT Examples – Build Tcl BI and Tclkits

Source directory

LibFileCopy
MultiFileCopy
Source directory: C:/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Install/tcludp/lib
Target directory: C:/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Development/opt/Tcl/lib
File pattern: *
Number of copied files: 2

LibFileCopy
MultiFileCopy
Source directory: C:/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Install/tcludp/lib
Target directory: C:/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Distribution/opt/Tcl/lib
File pattern: *
Number of copied files: 2

FileRename
Source: C:/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Development/opt/Tcl/lib/udp1.0.9
Target: C:/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Development/opt/Tcl/lib/udp

FileRename
Source: C:/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Distribution/opt/Tcl/lib/udp1.0.9
Target: C:/temp/Bawt-0.1.0/BawtBuild/Windows/x86/Release/Distribution/opt/Tcl/lib/udp

BAWT Stages

Bootstrap
Setup
Clean
Compile
Distribute
Finalize

Build directory

Action: Copy relevant files into developer and user distribution directories.
BAWT Examples – Build Tcl BI and Tclkits

**Source directory**

- Build file: C:/temp/Bawt-0.1.0/Bawt/Bawt-Tcl/TclRuntime.bawt
- Architecture: x86
- Compiler: Windows

**BAWT Stages**

- Clean
- Extract
- Configure
- Compile
- Distribute
- Finalize

**Build directory**

- Setup
- Clean
- Extract
- Configure
- Compile
- Distribute
- Finalize

**Summary**

### Build file

- C:/temp/Bawt-0.1.0/Bawt/Bawt-Tcl/TclRuntime.bawt

### Architecture

- x86

### Compiler

- Windows

### Stages

- Clean
- Extract
- Configure
- Compile
- Distribute
- Finalize

### # : Library Name | Version | Build time
--- | --- | ---
0: Tcl | 8.6.5 | 2.09 minutes
1: Tk | 8.6.5 | 2.50 minutes
2: TkImg | 1.4.6 | 16.62 minutes
3: tcludp | 1.0.9 | 0.66 minutes
4: tDOM | 0.8.3 | 0.93 minutes
5: Tktable | 2.10 | 0.91 minutes
6: tablelist | 5.15 | 0.04 minutes
7: Tclkit | 8.6.5 | 8.57 minutes
---

**Total:** 32.33 minutes

**Action:** Call user supplied `Finalize` procedure and print summary.
proc Build_tcludp { libName libVersion buildDir instDir devDir distDir } {
    set buildDirMSys [MSysPath $buildDir]
    set instDirMSys [MSysPath $instDir]
    set rootBuildDirMSys [MSysPath [GetBuildDir]]

    if { [UseStage "Extract"] } {
        ExtractLibrary $libName $buildDir
    }
    if { [UseStage "Configure"] } {
        set cmd ""
        append cmd "cd $buildDirMSys ; "
        append cmd "$buildDirMSys/configure "
        append cmd "--enable-shared "
        append cmd "--build=[GetMingwVersion] "
        append cmd "--prefix=$instDirMSys --exec-prefix=$instDirMSys "
        append cmd "--with-tcl=$rootBuildDirMSys/Tcl "
        if { [Is64Bit] } {
            append cmd "--enable-64bit "
        }
        if { [IsDebugBuild] } {
            append cmd "--enable-symbols "
        } else {
            append cmd "--disable-symbols "
        }

        MSysRun $libName "$cmd"
    }
    if { [UseStage "Compile"] } {
        MSysBuild $libName $buildDir "install-binaries"
    }
    if { [UseStage "Distribute"] } {
        LibFileCopy "$instDir" "$devDir/[GetTclDir]/" "*" true
        LibFileCopy "$instDir" "$distDir/[GetTclDir]/" "*" true

        FileRename "$devDir/[GetTclDir]/lib/udp$libVersion" "$devDir/[GetTclDir]/lib/udp"
        FileRename "$distDir/[GetTclDir]/lib/udp$libVersion" "$distDir/[GetTclDir]/lib/udp"
    }
}
### BAWT Examples – Build libraries for Tcl3D

```tcl
# Specify poSoft specific icon and resource file.
set thisDir [file dirname [GetBuildFile]]
SetTclkitIconFile [file join $thisDir "poSoft.ico"]
SetTclkitResourceFile [file join $thisDir "poSoft.rc"]

<table>
<thead>
<tr>
<th>Setup libName</th>
<th>zipFile</th>
<th>scriptFile</th>
<th>buildTypes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

# Tools needed for compilation of libraries.
if { [IsWindows] } {
  Setup CMakeWin32 CMakeWin32-3.5.2.7z CMakeWin32.bawt Release
} else {
  Setup CMake CMake-3.5.2.7z CMake.bawt Release
}

# Basic library needed by most other libraries.
Setup ZLib ZLib-1.2.8.7z ZLib.bawt Release

# Image libraries needed by OpenSceneGraph.
Setup GIF GIF-4.2.3.7z GIF.bawt Release
Setup JPEG JPEG-9.a.7z JPEG.bawt Release
Setup TIFF TIFF-4.0.3.7z TIFF.bawt Release
Setup PNG PNG-1.6.17.7z PNG.bawt Release

# Tcl and Tk.
Setup Tcl Tcl-8.6.5.7z Tcl.bawt Release
Setup TclStubs TclStubs-8.6.5.7z TclStubs.bawt Release
Setup Tk Tk-8.6.5.7z Tk.bawt Release
Setup TkStubs TkStubs-8.6.5.7z TkStubs.bawt Release

# Compiled Tcl packages.
...
# Compiled Tcl packages. Windows only.
Setup Twapi                   Twapi-4.2a3.7z               Twapi.bawt Release

# Pure Tcl packages.
Setup tablelist              tablelist-5.15.7z             tablelist.bawt Release
Setup Cawt                   Cawt-2.3a1.7z               Cawt.bawt Release

# Tclkits.
Setup Tclkit                  Tclkit-8.6.5.7z              Tclkit.bawt Release

# Tools needed for compilation of libraries.
Setup SWIG                    SWIG-3.0.5.7z               SWIG.bawt Release

# Tcl3D 3rd party libraries.
Setup GLEW                    GLEW-1.13.0.7z              GLEW.bawt Release
Setup Freetype                Freetype-2.4.4.7z             Freetype.bawt Release
Setup SDL                     SDL-2.0.3.7z                SDL.bawt Release
Setup FTGL                    FTGL-2.1.3.7z               FTGL.bawt Release
Setup Gl2ps                   Gl2ps-1.3.9.7z               Gl2ps.bawt Release

# OpenSceneGraph 3.0.1 does not compile on Darwin.
if { ! [IsDarwin] } { setup OpenSceneGraphData OpenSceneGraphData-3.0.1.7z OpenSceneGraphData.bawt Release
}
# BAWT Status – Overview of supported procedures

<table>
<thead>
<tr>
<th>Category</th>
<th>Supported Procedures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>CMakeConfig, MSysRun, DosRun</td>
</tr>
<tr>
<td>Compilation</td>
<td>CMakeBuild, NMakeBuild, MSBuild, MSysBuild, Dll2Lib</td>
</tr>
<tr>
<td>File handling</td>
<td>SingleFileCopy, MultiFileCopy, LibFileCopy, FileRename, DirCopy, DirDelete, MSysPath</td>
</tr>
<tr>
<td>Zip handling</td>
<td>Get7ZipProg, Unzip, ExtractLibrary</td>
</tr>
<tr>
<td>Build handling</td>
<td>UseStage, Setup, IsReleaseBuild, IsDebugBuild</td>
</tr>
<tr>
<td>Platform</td>
<td>IsWindows, IsLinux, IsDarwin, Is32Bit, Is64Bit</td>
</tr>
<tr>
<td>Environment variables</td>
<td>SetEnvVar, AddEnvVar, AddToPathEnv</td>
</tr>
</tbody>
</table>
BAWT Status – Build configuration

Usage: Build.bawt [Options] LibraryName [LibraryNameN]

General options:
--help : Print this help message and exit.
--loglevel <int>: Specify log message verbosity.
--check : Print all available library names and exit.

Build action options:
--clean : Clean library specific build and install directories.
--extract : Extract library sources from ZIP files or sub-directories.
--configure : Perform the configure stage of the build process.
--compile : Perform the compile stage of the build process.
--distribute: Perform the distribution stage of the build process.
--finalize : Generate environment file and call user supplied Finalize procedure.
--complete : Perform the following stages in order:
    clean, extract, configure, compile, distribution, finalize.

Build configuration options:
--architecture <string>: Build for specified processor architecture.
    Choices: x86, x64.
    Default: x86.
--compiler <string> : Build with specified compiler version.
    Default: Windows.
--gcc <string>      : Build with specified gcc version.
    Choices: 4.9.2, 5.2.0.
    Default: 4.9.2.
--buildtype <string> : Use specified build type.
    Choices: Release, Debug.
    Default: Specified in build configuration file.
--buildfile <string> : Specify build configuration file.
--numjobs <int>     : Number of parallel compile jobs.
    Default: 1
Prerequisites:
- None for building Tcl/Tk, TEA compliant packages or Tclkits.
- Otherwise Visual Studio or Visual Studio Express.
  - If VS is not installed in the standard location, you have to adapt procedure GetVcvarsProg.

Downloads:
- Bawt-0.1.0.zip (BAWT framework, appr. 5 MB)
- 7-zip.zip (Portable 7-Zip program, appr. 2 MB)
- MSys/MinGW distribution(s) (appr. 110 MB per distribution)

Actions:
- Extract BAWT-Framework Bawt-0.1.0.zip in a folder of choice, ex. C:\Bawt
- Copy 7-Zip.zip and MSys/MinGW distribution(s) into C:\Bawt\Bawt-0.1.0\Bootstrap-Windows
- Open command shell window and go into folder C:\Bawt\Bawt-0.1.0
- Create Tcl runtime distribution for 32-bit by typing:
  - Build-Windows.bat x86 vs2008 Bawt-Tcl\TclRuntime.bawt
- Create Tcl runtime distribution for 64-bit by typing:
  - Build-Windows.bat x86 vs2008 Bawt-Tcl\TclRuntime.bawt
BAWT Status – Build on Unix

Prerequisites Darwin:
- XCode
- curl (should be available by default on Mac)

Prerequisites Linux:
- C/C++ development
- Package p7zip (for 7z program)
- Package glu-devel (Optional: Needed by library Gl2ps)

Downloads:
- Bawt-0.1.0.zip (BAWT framework, appr. 5 MB)

Downloads Darwin:
- 7-zip.zip (command line version of 7-Zip, appr. 2 MB)

Actions:
- Extract BAWT-Framework Bawt-0.1.0.zip in a folder of choice.
- Open shell (Terminal window), go into created folder Bawt-0.1.0 and execute:
  - chmod u+x Build*
  - chmod u+x tclkit*
- Create Tcl runtime distribution for Darwin (only 64-bit) by typing:
  - ./Build-Darwin.sh Bawt-Tcl/TclRuntime.bawt
- Create Tcl runtime distribution for Linux (32-bit or 64-bit) by typing:
  - ./Build-Linux.sh x86 Bawt-Tcl/TclRuntime.bawt
  - ./Build-Linux.sh x64 Bawt-Tcl/TclRuntime.bawt
BAWT Status – What‘s on the TODO list

- Improve error checking and handling.
- Strip generated dynamic libraries in Release mode.
- Improve generation of debug libraries.
- More flexibility regarding source libraries (handling multiple versions).
- More flexibility regarding output directory structure.
- Add documentation.
- Add more Tcl extension packages.
- Add Tcl3D to BAWT build process.

Questions to the Tcl experts

- How to generate Tcl/Tk stubs libraries without using Visual Studio?
  - Currently using nmake/VS to compile Tcl/Tk source.

- How to generate sqlite3 and Co. without using Visual Studio?
  - There is a rm -Rf pkgs statement in configure script.

- How to generate import libs (*.lib) without using Visual Studio?
  - Currently using lib.exe from Visual Studio.
BAWT Summary

- BAWT allows the automatic building of C/C++ libraries and tools on Windows, Linux and Mac.
  - Works for my private Open Source projects.
  - Works for my projects at work.

- BAWT allows the automatic building of Tcl/Tk, Tcl extensions and Tclkits on Windows, Linux and Mac.
  - When using TEA compliant extensions, this is possible without the installation of a build environment.

Make your own Tcl Batteries Included

BAWT is available for testing at http://www.bawt.tcl3d.org

Special thanks to Stefan Wallner, who wrote a first version of BAWT implemented as DOS batch files.