C-Ware Software Toolset Application Guidelines

C-WARE SOFTWARE TOOLSET, VERSION 2.3

CSTOCAG-UG/D
Rev 01

CONTENTS

ABOUT THIS GUIDE

Guide Overview

This document presents requirements and guidelines for creating new C-Ware applications that conform to those of applicatons included in the C-Ware Software Toolset (CST).

This document is intended for software developers who create C-Ware-based applications using the CST.

This document also assumes good working knowledge of the C-Ware Software Toolset.

This document contains one chapter that contains the following major topics:

- Overview
- Directory Structure
- Naming Conventions
- Makefiles
- Requirements and Recommendations
- Checklist

Using PDF Documents

Electronic documents are provided as PDF files. Open and view them using the Adobe® Acrobat® Reader application, version 3.0 or later. If necessary, download the Acrobat Reader from the Adobe Systems, Inc. web site:

http://www.adobe.com/prodindex/acrobat/readstep.html

PDF files offer several ways for moving among the document's pages, as follows:

- To move quickly from section to section within the document, use the *Acrobat bookmarks* that appear on the left side of the Acrobat Reader window. The bookmarks provide an expandable 'outline' view of the document's contents. To display the

document's Acrobat bookmarks, press the 'Display both bookmarks and page' button on the Acrobat Reader tool bar.

- To move to the referenced page of an entry in the document's Contents or Index, click on the entry itself, each of which is "hot linked."
- To follow a cross-reference to a heading, figure, or table, click the blue text.
- To move to the beginning or end of the document, to move page by page within the document, or to navigate among the pages you displayed by clicking on hyperlinks, use the Acrobat Reader navigation buttons shown in this figure:

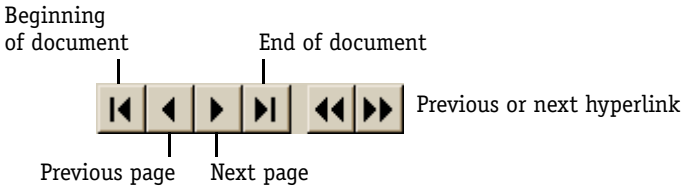

Table 1 summarizes how to navigate within an electronic document.

Table 1 Navigating Within a PDF Document

TO NAVIGATE THIS WAY	CLICK THIS
Move from section to section within the document.	A bookmark on the left side of the Acrobat Reader window
Move to an entry in the document's Contents or Index.	The entry itself
Follow a cross-reference (highlighted in blue text).	The cross-reference text
Move page by page.	The appropriate Acrobat Reader navigation buttons
Move to the beginning or end of the document.	The appropriate Acrobat Reader navigation buttons
Move backward or forward among a series of hyperlinks you have selected.	The appropriate Acrobat Reader navigation buttons

le Conventions

The following visual elements are used throughout this guide, where applicable:

This icon and text designates information of special note.

Warning: *This icon and text indicate a potentially dangerous procedure. Instructions contained in the warnings must be followed.*

Warning: *This icon and text indicate a procedure where the reader must take precautions regarding laser light.*

This icon and text indicate the possibility of electrostatic discharge (ESD) in a procedure that requires the reader to take the proper ESD precautions.

Revision History

Table 2 provides details about changes made for each revision of this guide.

Table 2 *C-Ware Reference Library Guide* Revision History

REVISION DATE	CST REVISION	CDS REVISION	CHANGES
April 12, 2002	2.1	2.0	Typographic corrections throughout.
September 11, 2001	2.0	2.0	New document.

Product ntation

Table 3 lists the documentation for the C-Ware library of reference applications.

Table 3 C-Ware Application Library Documentation Set

DOCUMENT NAME	PURPOSE	DOCUMENT ID
AAL-5 Fabric Port SAR to Gigabit Ethernet Switch Application Guide	Describes the key characteristics of the **gbeOc12SarFp** applications.	CSTAA5F2G-UG
AAL-5 SAR Application Guide	Describes the key characteristics of the **oc3SarQ** application.	CSTAA5-UG
AAL-5 SAR to Gigabit Ethernet Switch Application Guide	Describes the key characteristics of the **gbeOc12Sar** application.	CSTAA52G-UG
AAL-5 SAR to Gigabit Ethernet Switch with Q-5 TMC Application Guide	Describes the key characteristics of the **gbeOc12SarQ** application, which integrates the Q-5 TMC.	CSTAA52GQ-UG
ATM Cell Switch Application Guide	Describes the key characteristics of the **atmCellSwitchQ** application.	CSTAATMCS-UG
Differentiated Services Domain to IP Domain POS OC-12 Switch Application Guide	Describes the key characteristics of the **diffServPosOc12Q** application.	CSTDFSVQ-UG
FibreChannel to Gigabit Ethernet IP Gateway Application Guide	Describes the key characteristics of the **gbeFc** application.	CSTAFC2G-UG
Frame Relay to ATM to 10/100 Ethernet Switch Router Application Guide	Describes the key characteristics of the **switchRouter** application.	CSTAFRAE-UG
Gigabit Ethernet Switch Application Guide	Describes the key characteristics of the **gbeSwitch** application.	CSTAGBE-UG
Multi-PHY Switch Application Guide	Describes the key characteristics of the **mphySwitch** application.	CSTAMPHYS-UG
Packet Over SONET Switch Application Guide	Describes the key characteristics of the **posOc48Sc** application.	CSTAPOS-UG
Packet Over SONET to Ethernet Switch Application Guide	Describes the key characteristics of the **enetOc3Switch** application.	CSTAPOS2E-UG
Packet Over SONET to Gigabit Ethernet Switch Application Guide	Describes the key characteristics of the **posGbeSwitch** application.	CSTAPOS2G-UG
Voice Over IP to Voice Over ATM Media Gateway Application Guide	Describes the key characteristics of the **voIpToVoAtmSwitch** application.	CSTAVOIP-UG
Fabric Processor Configuration Component Guide	Describes the key characteristics of the **fabrics** application component.	CSTCFPC-UG
GMII Gigabit Ethernet Autonegotiation Component Guide	Describes the key characteristics of the **gmiiAutoNeg** application component.	CSTCGEAN-UG
ICMP Support Component Guide	Describes the key characteristics of the **ip** application component.	CSTCICMP-UG
MPC750 SBC Host Stack Support Component Guide	Describes the key characteristics of the **stackSupport** application component.	CSTCMHSS-UG

e 3 C-Ware Application Library Documentation Set (continued)

JMENT NAME	PURPOSE	DOCUMENT ID
PHY Configuration Component Guide	Describes the key characteristics of the **phy** application component.	CSTCPHYC-UG
QMU Configuration and RC Support Component Guide	Describes the key characteristics of the **queueUtils** application component.	CSTCQRCS-UG
SONET Monitoring Component Guide	Describes the key characteristics of the **sonet** application component.	CSTCSMC-UG
TLU Configuration Component Guide	Describes the key characteristics of the **tableUtils** application component.	CSTCTLUC-UG

Chapter 1

C-WARE SOFTWARE TOOLSET APPLICATION GUIDELINES

Overview

This document lists the requirements of new applications for C-Ware Software Toolset (CST) Version 2.0 and beyond. All applications that are categorized as Generally Available (GA) "product" applications must meet these requirements.

Product applications at the Alpha or Beta level may meet some of these requirements and must meet all requirements to be promoted to the GA level. Contrib applications may also meet some of these requirements.

This document also sets forth guidelines that are intended to help developers fit an application into the new directory structure and build system. The intent of the new directory structure and build system is to allow multiple targets to be built for each application in a flexible fashion.

At the end of the document, a checklist is provided to help developers identify whether an application meets the requirements of a GA Product.

Directory Structure

The directory structure defined in this section allows multiple targets to be built for an application and more importantly allows those targets to co-exist in a single development environment.

The C-Ware application build process, described in the *Build System Conventions* document, requires a consistent directory structure. Table 1 defines the directory structure to which all new applications must adhere. Directory names in <> indicate there may be multiple peer directories at this level.

Table 1 Directory Structure for C-Ware Applications

DIRECTORY	DESCRIPTION (EXAMPLE)
<module>	Top level module directory (gbeSwitch)
doc	Documentation (README, app design spec)
inc	Arch. independent interfaces

Table 1 Directory Structure for C-Ware Applications (continued)

DIRECTORY	DESCRIPTION (EXAMPLE)
src	Arch. independent source code
run	Simulation files (config, sim.in)
inPatterns	Input pattern files and pattern generation makefile
outPatterns	Output pattern files captured during simulation
tgt	Required directory as token for build tool
<family>	Architecture (c5, q5, m5, host)
inc	Processor independent interfaces
src	Processor independent source code
<proc>	Processor (xprc, cprc, sdp, fdp)
inc	Chip independent interfaces
src	Chip independent source code
<chip>	Chip (c5i, c5ii)
inc	Chip revision independent interfaces
src	Chip revision independent source code
<rev>	Chip revision (a0, b0, etc.)
inc	Chip revision dependent interfaces
src	Chip revision dependent source code

ıing Conventions

File names are defined as a concatenation of fields and optional fields.

- Source code (**.c** files):

  `<func>[<sub-func>][<proc>[<phase>]].c`

- Headers (**.h** files):

  `<func>[<sub-func>][{<proc>|If}].h`

FIELD	DESCRIPTION
<func>	Primary function of the module, often a protocol (for example, mac, ip, enet).
<sub-func>	Optional sub-function. This is useful when the function described by <func> is better maintained by splitting it into pieces (for example, Rx, Tx).
<proc>	Intended processor for this code (for example, Cp, Xp).
<phase>	Phase of a multi-phase program (for example, Main, Init). If this field is present, the <func> field is not required (for example, **xpMain.c**).

Some valid examples would be: **mac.h**, **atmXp.c**, **iplf.h**, **sonetMon.c**, **sarCpMain.c**, and **xpMainInit.c** (in this case 'Main' and 'Init' is the <phase>).

The only exception to this rule are makefiles which are named "Makefile", with a capital M.

es

The application makefile will include several standard targets. These targets are necessary so that a parent makefile will be able to invoke a child makefile in a consistent manner. Table 2 summarizes these targets.

Table 2 Standard Makefile Targets

TARGET NAME	DESCRIPTION
default	First target in the makefile. Has a single dependency of "all" and no rules. Invoking make on the command line with no targets will build the target "all."
all	Builds all necessary code images. (e.g. the *.pkg file)
patterns	Generates all input pattern files used for regression testing.
test	Runs the regression test.
accept	Builds the code image, generates the patterns, and runs the regression test. Equivalent to "make all patterns accept."
clean	Removes all files created by the make process from all of the standard targets.

Also note that Makefiles must support GNU emacs TAGS.

Here is a portion of an example makefile with illustrating the use of the standard targets:

```
default:          all

all:             $(TGT)/widget.pkg

patterns:    $(INPAT_DIR)/widgetIn.pat
                cd $(INPAT_DIR) && $(MAKE)

test:
                $(DCPSIM) –batch
                .
                .
                .

accept:       all patterns test
```

```
clean:
        rm -r $(TGT)
        rm -f $(INPAT_DIR)/*.pat
        .
        .
        .
```

Requirements and Recommendations

The following table lists a set of requirements and recommendations on applications that are a part of the CST.

These are split out into the following categories:

- Applications' requirements
- Naming conventions
- Documentation guidelines
- Miscellaneous guidelines
- Test guidelines

Applications' Requirements

Table 3 enumerates the requirements of applications in the CST.

Table 3 Requirements of CST Applications

DESCRIPTION	REQUIRED
The directory structure will meet the specifications in Section.	Yes
The only leaf directories at the <family> level or below will be "inc" and "src".	Yes
The only leaf directories at the <module> level will be "inc", "src", "doc", and "sim".	Yes
The "sim" directory may contain a "patterns" directory.	No
Source code should be placed in the directory tree at the highest level as practical.	No
As a source code file becomes more target specific, as in the case of a new chip being added to the directory structure, it should be demoted down the directory tree as appropriate.	No
Source code which is expected to be portable across all (or many) architectures should be placed in the "<module>/src" directory and its public interface should be placed in the "<module>/inc" directory.	No

Table 3 Requirements of CST Applications (continued)

DESCRIPTION	REQUIRED
Source code which implements interfaces between processors in a single architecture should be placed in the "…/<family>/src" directory and its public interface should be placed in the "…/<family>/inc" directory.	No
Source code which is intended to run on a particular processor should be placed in the "…/<proc>/src" directory and its public interface should be placed in the "…/<proc>/inc" directory.	No
Source code which includes chip specific implementation should be placed in the "…/<chip>/src" directory and its public interface should be placed in the "…/<chip>/inc" directory.	No
Source code which included chip revision specific implementation should be place in the "…/<rev>/src" directory and its public interface should be placed in the "…/<rev>/inc" directory.	No

Naming Conventions

Table 4 enumerates the documentation requirements of CST applications.

Table 4 Naming Conventions Requirements of CST Applications

DESCRIPTION	REQUIRED
All files and directories will be named using a mixed case convention where the first letter is always lowercase and the beginning of each subsequent logical word is uppercase.	Yes
Filenames will meet the specifications in Section.	Yes
Makefiles will be named "Makefile"	Yes
Simulation configuration files will be named "config[<desc>]".	Yes
Simulation input files will be named "sim[<desc>].in".	Yes
Filenames should not contain underscores.	No
Header files used to export public functions and data should have the same file name root as the source file in which the functions and data are contained.	No
Header files that describe interfaces between physical components should have a file name root that ends in "If".	No

umentation Guidelines

Table 5 enumerates the documentation requirements of CST applications.

Table 5 Documentation Requirements of CST Applications

DESCRIPTION	REQUIRED
A README file, described in a separate document, will be included in the "<module>/doc" directory of each application or component and is supplied with the CST.	Yes
An application design specification, described in a separate document, will be included in the "<module>/doc" directory of each application and is intended for internal use.	Yes
The application design specification will be made available to the Technical Publications group for use in the CST application user's guide.	Yes
Outstanding issues, known limitations, and incomplete features will be listed in the CST release notes.	Yes

Test Guidelines

Table 6 enumerates the test guidelines for applications in the CST.

Table 6 Test Guidelines for CST Applications

DESCRIPTION	REQUIRED
A regression test will be provided that runs under simulation and that can be invoked from the application's makefile.	Yes
A test procedure should be provided if the application runs on hardware.	No

Miscellaneous Guidelines

Table 7 enumerates the miscellaneous guidelines for applications in the CST.

Table 7 Miscellaneous Guidelines for CST Applications

DESCRIPTION	REQUIRED
A design review will be held for each new application.	Yes
The application will compile and link with no errors or warnings.	Yes
The application makefile will contain the targets described in Section.	Yes

st

Table 8 presents a checklist to aid developers in making sure that an application meets all the requirements of the CST.

Table 8 Checklist of Requirements of CST Applications

DESCRIPTION	COMPLETE?
proper directory structure	
properly named source code and header files	
properly named makefile and simulation files	
README file	
design specification complete and submitted to tech pubs	
input for release notes submitted to tech pubs	
regression test	
design review	
no errors or warnings during build	
standard targets in makefile	
support for GNU emacs TAGS	