ChorusOS 4.0 MPC8260 Target Family Guide

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A.

Part Number 806-3965–10
December 1999

Contents

CHAPTER 1

ChorusOS 4.0 MPC8260 Target Family Guide

This guide describes how to run the ChorusOS™ 4.0 product for the MPC8260 processor family.

Preface

How This Guide is Organized

ChorusOS 4.0 MPC8260 specific information is provided in the following major sections:

- "Development Environment" on page 8, includes supported hosts, host operating systems and development systems.
- "ChorusOS 4.0 Supported Features" on page 9, includes kernel components and POSIX components.
- "Libraries" on page 12.
- "Utilities" on page 13, includes host and target utilities.
- "Reference Hardware" on page 16, includes supported reference platforms, supported devices, and validated reference platforms.
- "How to Build and Boot a System Image on the Target" on page 18.
- Appendix A, details the list of Solaris packages in the product components, and the associated part numbers.

Related Books

See the *ChorusOS 4.0 Installation Guide for Solaris Hosts* for a description of the installation process of the ChorusOS 4.0 product on a host workstation running the Solaris™ operating environment. This document also describes how to set up a boot server running the Solaris operating environment.

See the *ChorusOS 4.0 Introduction* for a complete description of the ChorusOS 4.0 features.

Typographical Conventions

The following table describes the typographic changes used in this book.

TABLE 1–1 Typographical Conventions

Typeface or Symbol	**Meaning**	**Example**
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls –a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. You must be *root* to do this.

Shell Prompts

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE 1–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

Ordering Sun Documents

Fatbrain.com, an Internet professional bookstore, stocks selected product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at `http://www1.fatbrain.com/documentation/sun`.

Accessing Sun Documentation Online

The docs.sun.com[SM] Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Obtaining Technical Support

Sun Support Access offerings are available exclusively to members of the Sun Developer Connection Program. To get free membership in the Sun Developer Connection Program, go to `http://www.sun.com/developers`. For more information or to purchase Sun Support Access offerings, visit: `http://www.sun.com/developers/support` or contact the Sun Developer Connection Program office near you.

Development Environment

The ChorusOS 4.0 product provides a host-target development environment. Applications are developed on a workstation (the host), and then downloaded and executed on a specific board (the target).

A cross development system is needed to build the applications that execute on the target board (see Section “Utilities” on page 13).

SPARC™ / Solaris™ Reference Host Environments

Prerequisites for the Solaris host reference configuration are the following:

- Sun SPARCstation™
- Solaris 2.6, or Solaris 7
- JDK™ 1.1.3 to 1.1.8, for the installation tool
- JDK 1.2, for the graphical configuration tool

Cross Compiler

This development environment component is bundled with the ChorusOS 4.0 for MPC8260 product:

- Chorus Cross Development System 5.0, target PowerPC ELF

The Chorus Cross Development System is based on the Experimental GNU Compiler System egcs 1.1.2 and binutils 2.9.1 and additional patches.

Graphical Debugger

This development environment component is bundled with the ChorusOS 4.0 for MPC8260 product:

- XRAY Debugger from Mentor Graphics, target PowerPC ELF version 4.4crb and additional patches.

ChorusOS 4.0 Supported Features

The following table shows the ChorusOS kernel and operating system optional features that are available for the MPC8260 processor family. The availability status of a feature, can be one of:

Y The feature is supported, and is configurable with the `configurator`(1CC) command, or with the *ews* GUI configuration tool.

Please refer to the note at the end of the table for information about specific conditions, or restrictions, for a given supported feature.

Some of the features (such as MSDOSFS, FLASH, FS_MAPPER, for example) require specific low-level drivers. These features operate only on platforms which provide these drivers.

N The feature is not supported.

Feature Description	Feature Name	Availability
Actor management		
Dynamic actor loading management	ACTOR_EXTENDED_MNGT	Y
User-mode extension support	USER_MODE	Y
Dynamic libraries	DYNAMIC_LIB	Y [1]
Compressed file management	GZ_FILE	Y
Scheduling		
POSIX round-robin scheduling class	ROUND_ROBIN	Y
Memory management		
Virtual (user and supervisor) address space	VIRTUAL_ADDRESS_SPACE	Y [2]
On-demand paging	ON_DEMAND_PAGING	Y [2]
Hot restart and persistent memory		
Hot restart	HOT_RESTART	Y
Inter-thread communication		

Feature Description	Feature Name	Availability
Semaphores	SEM	Y
Event flag sets	EVENT	Y
Mutual exclusion lock supporting thread priority inversion avoidance	RTMUTEX	Y
Time management		
Periodic timers	TIMER	Y
Thread and actor virtual timer	VTIMER	Y
Date and time of day	DATE	Y
Real-time clock	RTC	Y
Inter-process communication		
Location-transparent inter-process communication	IPC	Y
Remote (inter-site) IPC support	IPC_REMOTE	Y
Remote IPC communications medium	IPC_REMOTE_COMM	Y
Mailbox-based communications mechanism	MIPC	Y
POSIX 1003.1-compliant message queues	POSIX_MQ	Y
POSIX 1003.1-compliant shared memory objects	POSIX_SHM	Y
LAP		
Local name server for LAP binding	LAPBIND	Y
LAP validity-check option	LAPSAFE	Y
Tools support		
Message logging	LOG	Y
Profiling and benchmark support	PERF	Y
System Monitoring	MON	Y
System debugging	DEBUG_SYSTEM [3]	Y
C_INIT		
Basic command interpreter on target	LOCAL_CONSOLE	Y

Feature Description	Feature Name	Availability
Remote shell	RSH	Y
File system options		
Named pipes	FIFOFS	Y
MS-DOS file system	MSDOSFS	Y
NFS client	NFS_CLIENT	Y
NFS server	NFS_SERVER	Y
UFS file system	UFS	Y
I/O management		
Network packet filter	BPF	Y
Swap support	FS_MAPPER	Y
Driver for IDE disk	IDE_DISK	N
`/dev/mem`, `/dev/kmem`, `/dev/null`, `/dev/zero`	DEV_MEM	Y
Support for RAM disk	RAM_DISK	Y
Support for FLASH media	FLASH	Y
Virtual TTY	VTTY	Y
Driver for SCSI disk	SCSI_DISK	N
Support for IPC	IOM_IPC	Y
Support for OSI	IOM_OSI	Y
Networking		
Serial link IP	SLIP	Y
POSIX 1003.1g-compliant sockets	POSIX_SOCKETS	Y
Point-to-point protocols	PPP	Y
Local sockets and pipes	AF_LOCAL	Y
Administration		
ChorusOS statistics	ADMIN_CHORUSSTAT	Y
`ifconfig` administration command	ADMIN_IFCONFIG	Y
`mount` administration command	ADMIN_MOUNT	Y
`rarp` administration command	ADMIN_RARP	Y
`route` administration command	ADMIN_ROUTE	Y

Feature Description	Feature Name	Availability
`shutdown` administration command	ADMIN_SHUTDOWN	Y
`netstat` administration command	ADMIN_NETSTAT	Y

1. Limitation: the binaries making up the executing image of an actor (main program and dynamic libraries) must hold in a 32MB address range. Even if their total size is less than 32 MB, this is not garanteed in flat mode or for supervisor actors.
2. If the value for VIRTUAL_ADDRESS_SPACE is `true`, the value for ON_DEMAND_PAGING is `true`. If the value for VIRTUAL_ADDRESS_SPACE is `false`, the value for ON_DEMAND_PAGING is `false`.
3. A flashed system image configured with DEBUG_SYSTEM enabled does not boot. The DEBUG_SYSTEM feature must be disabled.

Libraries

The ChorusOS operating system provides the elementary libraries indicated in the following list:

ChorusOS embedded library [1]	`libebd.a`
ChorusOS extended library [1]	`libcx.a`
C++ library	`libC.a`
X11 related client libraries (not thread safe)	`libX11.a`, `libXaw.a`, `libXext.a`, `libXmu.a`, `libXt.a`
Specific BSD APIs (not thread safe)	`libbsd.a`
The SunRPC library	`librpc.a`
The mathematical library	`libm.a`

The ‘‘embedded’’ C library [2]	`stdc.a`
The microkernel ‘‘visu’’ library [3]	`visu.a`

1. The `libebd.a`, `libcx.a`, `libm.a` and `libC.a` libraries have been made thread-safe in order to support multithreaded actors.
2. Included in `libebd.a`
3. This library is provided for the sake of backwards compatibility only. It is not documented. Its use is strongly discouraged.

Utilities

Target Utilities

The following utilities may be run on the target ChorusOS operating system:

chorusStat(1CC)

cp(1CC)

cs(1CC)

date(1CC)

dd(1CC)

df(1CC)

domainname(1CC)

ftp(1CC)

hostname(1CC)

ls(1CC)

mkdir(1CC)

mkfifo(1CC)

mv(1CC)

netstat(1CC)

nfsstat(1CC)

pax(1CC)

PROF(1CC)

profctl(1CC)

rdbc(1CC)

rm(1CC)

rmdir(1CC)

touch(1CC)

uname(1CC)

ypcat(1CC)

ypmatch(1CC)

ypwhich(1CC)

arp(1M)

chat(1M)

chorusNS(1M)

chorusNSinet(1M)

chorusNSsite(1M)

dhclient(1M)

disklabel(1M)

flashdefrag(1M)

format(1M)

fsck(1M)

fsck_dos(1M)

ftpd(1M)

inetNS(1M)

inetNSdns(1M)

inetNShost(1M)

inetNSien116(1M)

inetNSnis(1M)

mkfd(1M)

mkfs(1M)

mount(1M)

mount_msdos(1M)

mount_nfs(1M)

mountd(1M)

newfs(1M)

newfs_dos(1M)

nfsd(1M)

portmap(1M)

shutdown(1M)

slattach(1M)

syncd(1M)

sysctl(1M)

telnetd(1M)

umount(1M)

ypbind(1M)

Host Utilities

The following utilities may be run on the host machine:

chadmin(1CC)

chconsole(1CC)

chlog(1CC)

chls(1CC)

ChorusOSMkMf(1CC)

chserver(1CC)

configurator(1CC)

configure(1CC)

ews(1CC)

mkmerge(1CC)

rdbs(1CC)

profrpg(1CC)

Reference Hardware

ChorusOS 4.0 targets are described in this section from three different points of view:

Reference Processors and BSPs:

This subsection describes the processors on which the ChorusOS 4.0 product can run as well as the details of the BSPs included in the delivery

Reference Target Platforms:

This section describes all the target platforms which can be used as references in the context of Sun support contracts

Validated Reference Targets:

This section describes the precise platforms used to run the Sun QA tests; this may be useful, in case of bugs, as a hint or guide to help in identifying issues which are closely hardware related.

Reference Processors and BSPs

The ChorusOS 4.0 system for MPC8260 supports the following processor:

- Motorola PowerPC 8260

The ChorusOS 4.0 system for MPC8260 supports the following reference BSP:

- sbc8260 Reference BSP

sbc8260 Reference BSP

Systems

The sbc8260 reference BSP supports the following board:

SBC8260 – EST corp

Devices

The sbc8260 reference BSP supports the following on board devices:

Device Id	ChorusOS Driver
/cpu (time base and decrementer)	sun:powerpc-(tb,dec)-timer
/flash (FLASH memory)	not supported
/quicc8260/ (Quicc bridge)	sun:powerpc-mpc8260-quicc
/quicc8260/smc-2 (Uart)	sun:quicc-smc-uart
/quicc8260/smc-1 (Uart) [1]	sun:quicc-smc-uart
/quicc8260/scc-1 (Ethernet) [2]	not supported
/quicc8260/fcc-2 (Ethernet)	sun:quicc-fcc-ether

1. smc-1 is normally used for the console. In that case it cannot be used for ppp.
2. As of the date of publication of this guide, the processor provided by Motorola does not support ethernet on scc-1 (See the MPC8260 Device errata dated September 30 1999 #CPM34). The device has been tested with the sun:quicc-scc-ether driver (available on MPC8xx), and it is possible to send and receive small packets.

Reference Target Platforms

Reference target platforms are configurations to be used by customers covered by a Sun support contract.

SBC8260 (EST)

Type: Evaluation/Development Board

Processor: MPC8260 PowerQUICC II (132 Mhz Core, 66 Mhz CPM)

Main memory: 16-32 MB

Devices: Asynchronous serial ports (38.4 Kbaud), 10BaseT Ethernet, Timers

Host debugger: visionXD (via visionICE emulator)

Validated Reference Targets

This section describes the precise platform used to run the Sun QA tests

- SBC8260: PCB-00089 Rev.02

How to Build and Boot a System Image on the Target

For the following procedures, it is assumed that you use the EST Corporation's visionXD tool either to boot the target system or to place the bootMonitor utility in flash memory on the target system. As of the date of publication of this document, the URL for the visionXD tool is: `http://www.estc.com/products/visionXD/index.html`

▼ Building a ChorusOS 4.0 System Image

The following procedure assumes that the ChorusOS 4.0 product has already been correctly installed on the host workstation. See the *ChorusOS 4.0 Installation Guide for Solaris Hosts*.

1. Create and change to a build directory where you will build system images:

```
$ mkdir build_dir
$ cd build_dir
```

2. Set an environment variable to use with the `configure`(1CC) command as a shortcut to the base directory.

For example:

Set the environment variable…	To the family-specific product directory. The default value is…
DIR	/opt/SUNWconn/SEW/4.0/chorus-mpc8260

3. **Make sure your PATH has been set correctly to include the directory** *install_dir*`/4.0/chorus-mpc8260/tools/host/bin`**, where the default** *install_dir* **is** `/opt/SUNWconn/SEW`**. Also make sure that your PATH includes** `/usr/openwin/bin`**, which contains the** `imake` **utility.**

4. **Configure the build directory, using the** `configure`**(1CC) command**:

   If you are building from a binary distribution:

```
$ configure -b $DIR/kernel \
$DIR/os \
$DIR/tools \
-s $DIR/src/nucleus/bsp/drv \
$DIR/src/nucleus/bsp/powerpc \
$DIR/src/nucleus/bsp/powerpc/mpc8260ADS \
$DIR/src/iom
```

Note - The above command configures the build directory to include components installed during a “Default Install”. It does not include optional components, such as the X library or code examples, that you may choose to install separately on Solaris host workstations. For example, in order to include everything in your build environment:

```
$ configure -b $DIR/kernel \
$DIR/os \
$DIR/opt/X11 \
$DIR/tools \
-s $DIR/src/nucleus/bsp/drv \
$DIR/src/nucleus/bsp/powerpc \
$DIR/src/nucleus/bsp/powerpc/mpc8260ADS \
$DIR/src/iom \
$DIR/src/opt/examples
```

If you are building from the source distribution, see the *ChorusOS 4.0 Production Guide*.
As a result of configuration, *build_dir* now contains a `Makefile`, which is used to generate the build environment, and a `Paths` file, which specifies paths to files required by and created in the build environment.

5. **Generate the build environment:**

```
$ make
```

6. **Build a system image:**

```
$ make chorus
```

The resulting system image file is located in the build directory, *build_dir* and is called `chorus.RAM`.

Note - You can also make a smaller system image that includes only the operating system kernel:

```
$ make kernonly
```

▼ Setting Up the visionXD and visionICE Tools

See the visionXD and visionICE documentation for detailed installation instructions.

1. **Connect** visionICE **to the target system, plugging the transceiver into the** MII **connector and switching the transceiver to** UP**.**
2. **Connect the target system to the network.**
3. **Connect and configure** visionICE **and** visionNET**.**
4. **Install the** visionXD **tool on the host workstation.**
5. **Select** Configure Global Settings... **from the** Tasks **menu, and use the dialog box displayed to configure the connection to the target system.**

How to Boot the Target System Using EST Corporation's visionXD Tool

▼ Booting with the visionXD Tool

1. **Start the** visionXD **tool.**
2. **Select** Connect **from the** Target **menu to connect the** visionXD **tool to the target system.**
3. **Click the terminal button to display a terminal window with the** >BKM> **prompt.**
4. **Enter** in **at the** >BKM> **prompt to set initialization values for the target system registers:**

```
>BKM> in
```

5. **Select** Load Executable... **from the** File **menu.**
6. **Select the** chorus.RAM **system image and click** Load**.**

7. **After the target system has finished downloading the system image, click the** Run **button.**

How to Boot the Target System from Flash Memory Using bootMonitor

In order to boot the target from flash memory you must perform the following procedures.

▼ Placing the System Image on the Boot Server

See the *ChorusOS 4.0 Installation Guide for Solaris Hosts* for instructions on how to configure the boot server.

1. **Copy the system image to the boot server.**

   For example, on a Solaris host workstation:

```
$ rcp chorus.RAM boot_server:/tftpboot
```

2. **Verify that everyone has at least read access to the system image on the boot server.**

   For example:

```
$ rlogin boot_server
Password: password_for_user
$ ls -l /tftpboot/chorus.RAM
-rwxr-xr-x   1 user    group     1613824 Dec 15 17:33 chorus.RAM*
```

3. **While logged in to the boot server, create a configuration file for the target.**

   For a target system with IP address `129.157.173.199` using a boot server with IP address `129.157.173.144`, the configuration file contains the following:

```
AUTOBOOT=YES
BOOTFILE=chorus.RAM
BOOTSERVER=129.157.173.144
```

The configuration file is named `/tftpboot/819DADC7.ChorusOS.4.0`, which is constructed from the target system IP address `129.157.173.199` as a concatenation of the following:

- `129` in decimal translates to `81` in hexadecimal
- `157` in decimal translates to `9D` in hexadecimal
- `173` in decimal translates to `AD` in hexadecimal
- `199` in decimal translates to `C7` in hexadecimal
- (optional) `.ChorusOS.4.0` identifies the release, and is appended to the concatenation of the IP address expressed in hexadecimal.

Note - The system first attempts to find the configuration file with the `.ChorusOS.4.0` extension. If it fails to find one, however, it attempts to find a configuration file without the `.ChorusOS.4.0` extension.

▼ Creating a bootMonitor Image

See `bootMonitor`(1CC) for details about how bootMonitor works.

1. **Create a build directory where you will build a** bootMonitor **image:**

```
$ mkdir bootmon
$ cd bootmon
```

Note that this build directory is different from the directory where you build system images.

2. **Configure the** bootMonitor **build directory based on the binary distribution:**

```
$ configure -b $DIR/kernel \
$DIR/os \
$DIR/tools \
-s $DIR/src/nucleus/bsp/drv \
$DIR/src/nucleus/bsp/powerpc \
$DIR/src/nucleus/bsp/powerpc/mcp8260ADS \
```

(continued)

```
$DIR/src/iom
```

3. Generate the build environment:

```
$ make
```

4. Edit the special *bootmon*/conf/mini profile so that it reads:

```
#
# Mini Profile
#

#
# Kernel features
#
-set USER_MODE=false
-set VIRTUAL_ADDRESS_SPACE=false
-set SEM=false
-set EVENT=false
-set MONITOR=false
-set TIMER=false
-set DATE=false
-set RTC=false
-set PERF=false
-set IPC=false
-set MIPC=false
-set LAPBIND=true # Change this from 'false' to 'true'
-set LAPSAFE=true # Change this from 'false' to 'true'
-set MON=false
-set LOG=false
```

5. Configure the build environment for bootMonitor**:**

```
$ configurator -p conf/mini
$ configurator -set BOOT_MODE=ROM
$ configurator -set ETHER_ADDR=xx:xx:xx:xx:xx:xx
```

As you enter the commands above, replace *xx*:*xx*:*xx*:*xx*:*xx*:*xx* with the target system Ethernet address.

6. **Build a** bootMonitor **image:**

```
$ make bootMonitor
```

The resulting system image file is located in the build directory, *bootmon* and is called `bootMonitor.ROM`.

▼ Flashing the Target System with the bootMonitor Image

1. **Start the** visionXD **tool.**

2. **Click the terminal button to display a terminal window with the** >BKM> **prompt.**

3. **Enter** `in` **at the** >BKM> **prompt to set initialization values for the target system registers.**

```
>BKM> in
```

4. **Enter** `cs` **at the** >BKM> **prompt to check that the SDRAM configuration as coded in the** `trampoline.s` **source file used to build the** bootMonitor **image corresponds to the initialization values set using the** `in` **command.**

```
>BKM> cs
```

5. **Select** `Program Flash...` **from the** `Tasks` **menu.**

The `Flash Programmer` window is displayed. Use the information in the table below to fill in the necessary fields in the window.

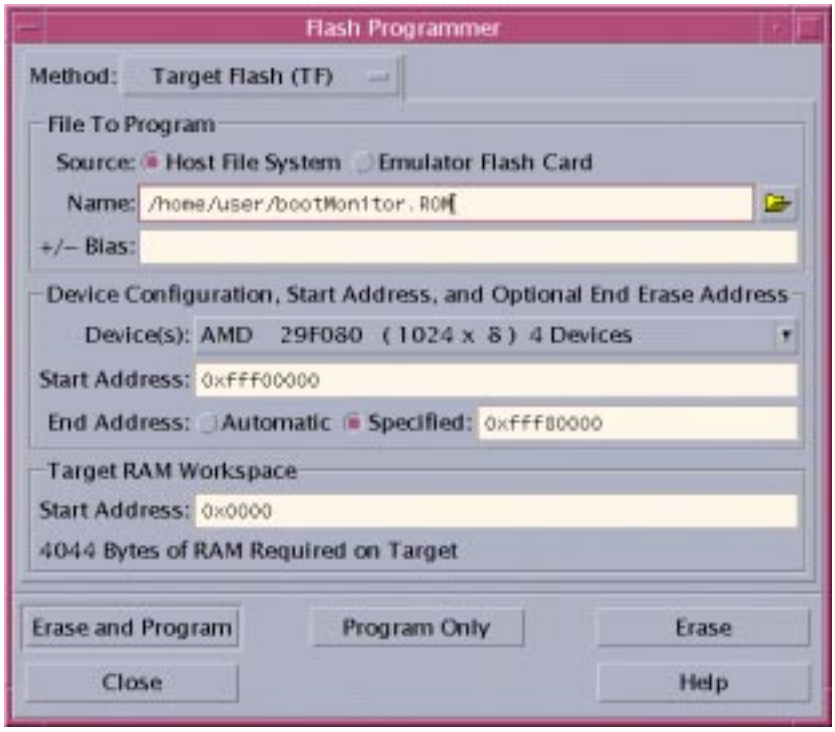

Figure 1–1 Configuring the Flash Programmer

Click the `Erase and Program` button to write the bootMonitor image to flash.

6. **Click `Extract` in the dialog box that is displayed.**

7. **Click `OK` to confirm the download into flash memory.**

▼ Booting the Target System

- **Restart the target system to boot from flash.**

APPENDIX **A**

ChorusOS 4.0 for MPC8260 Product Packages and Part Numbers

The tables below list the Solaris packages available in this release and indicate the part number for each distinct product component.

Binary Product — for Solaris Host

Part Number	**CLX400-SG90**
Package Name	**Description**
SUNWewbq	Sun Embedded Workshop for MPC8260 BSP source
SUNWewcd	Sun Embedded Workshop PDF Format Common Documentation
SUNWewch	Sun Embedded Workshop HTML Format Common Documentation
SUNWewcp	Sun Embedded Workshop PostScript Format Common Documentation
SUNWewdq	Sun Embedded Workshop for MPC8260 XRAY Debugger
SUNWewgq	Sun Embedded Workshop for MPC8260 GUI Tools
SUNWewiq	Sun Embedded Workshop for MPC8260 IOM source
SUNWewkq	Sun Embedded Workshop for MPC8260 Kernel

Part Number	CLX400-SG90
Package Name	**Description**
SUNWewm	Sun Embedded Workshop On-Line Manual Pages
SUNWewoq	Sun Embedded Workshop for MPC8260 OS
SUNWewpq	Sun Embedded Workshop for MPC8260 Examples
SUNWewsd	Sun Embedded Workshop PDF Format Specific Documentation
SUNWewsh	Sun Embedded Workshop HTML Format Specific Documentation
SUNWewsp	Sun Embedded Workshop PostScript Format Specific Documentation
SUNWewtq	Sun Embedded Workshop for MPC8260 Build Tools
SUNWewuq	Sun Embedded Workshop for MPC8260 Debugger and Profiling Support
SUNWewxq	Sun Embedded Workshop for MPC8260 X11 Library
SUNWewzq	Sun Embedded Workshop for MPC8260 egcs Toolchain

Flite Add-on for Solaris Host

Part Number	FLT400-SG90
Package Name	**Description**
SUNWewfq	Sun Embedded Workshop for MPC8260 Flite

Source Add-on for Solaris Host

Part Number	**CLX400-SG90-S**
Package Name	**Description**
SUNWewhq	Sun Embedded Workshop for MPC8260 OS source
SUNWewlq	Sun Embedded Workshop for MPC8260 Kernel source

Documentation for Solaris Host

Part Number	**CLX400-SAA0-D1N**
Package Name	**Description**
SUNWewcd	Sun Embedded Workshop PDF Format Common Documentation
SUNWewch	Sun Embedded Workshop HTML Format Common Documentation
SUNWewcp	Sun Embedded Workshop PostScript Format Common Documentation
SUNWewm	Sun Embedded Workshop On-Line Manual Pages
SUNWewsd	Sun Embedded Workshop PDF Format Specific Documentation
SUNWewsh	Sun Embedded Workshop HTML Format Specific Documentation
SUNWewsp	Sun Embedded Workshop PostScript Format Specific Documentation