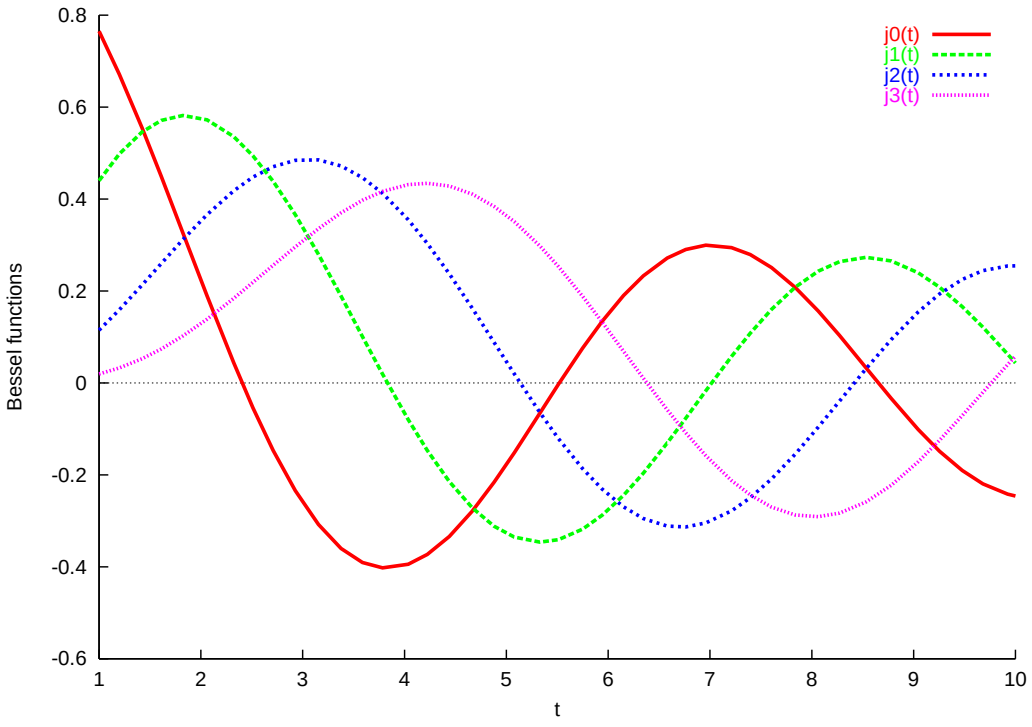
SoftIntegration®

The Ch Language Environment

Version 6.1

Installation and System Administration Guide



How to Contact SoftIntegration

Mail	SoftIntegration, Inc. 216 F Street, #68 Davis, CA 95616
Phone	+ 1 530 297 7398
Fax	+ 1 530 297 7392
Web	http://www.softintegration.com
Email	info@softintegration.com

Copyright ©2001-2008 by SoftIntegration, Inc. All rights reserved.

Revision 6.1.0, September 2008

Permission is granted for registered users to make one copy for their own personal use. Further reproduction, or any copying of machine-readable files (including this one) to any server computer, is strictly prohibited.

SoftIntegration, Inc. is the holder of the copyright to the Ch language environment described in this document, including without limitation such aspects of the system as its code, structure, sequence, organization, programming language, header files, function and command files, object modules, static and dynamic loaded libraries of object modules, compilation of command and library names, interface with other languages and object modules of static and dynamic libraries. Use of the system unless pursuant to the terms of a license granted by SoftIntegration or as otherwise authorized by law is an infringement of the copyright.

SoftIntegration, Inc. makes no representations, expressed or implied, with respect to this documentation, or the software it describes, including without limitations, any implied warranty merchantability or fitness for a particular purpose, all of which are expressly disclaimed. Users should be aware that included in the terms and conditions under which SoftIntegration is willing to license the Ch language environment as a provision that SoftIntegration, and their distribution licensees, distributors and dealers shall in no event be liable for any indirect, incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of the Ch language environment, and that liability for direct damages shall be limited to the amount of purchase price paid for the Ch language environment.

In addition to the foregoing, users should recognize that all complex software systems and their documentation contain errors and omissions. SoftIntegration shall not be responsible under any circumstances for providing information on or corrections to errors and omissions discovered at any time in this documentation or the software it describes, even if SoftIntegration has been advised of the errors or omissions. The Ch language environment is not designed or licensed for use in the on-line control of aircraft, air traffic, or navigation or aircraft communications; or for use in the design, construction, operation or maintenance of any nuclear facility.

Ch, SoftIntegration, and One Language for All are either registered trademarks or trademarks of SoftIntegration, Inc. in the United States and/or other countries. Microsoft, MS-DOS, Windows, Windows 95, Windows 98, Windows Me, Windows NT, Windows 2000, and Windows XP are trademarks of Microsoft Corporation. Solaris and Sun are trademarks of Sun Microsystems, Inc. Unix is a trademark of the Open Group. HP-UX is either a registered trademark or a trademark of Hewlett-Packard Co. Linux is a trademark of Linus Torvalds. Mac OS X and Darwin are trademarks of Apple Computers, Inc. QNX is a trademark of QNX Software Systems. All other trademarks belong to their respective holders.

Table of Contents

Chapter 1

Introduction

1.1 Welcome to Ch

Ch (pronounced C H) is a C-compatible cross-platform language environment. Ch is also a C virtual machine and a superset of C interpreter with salient features from C++, other languages and software packages. Ch provides an elegant way for programmers, system administrators, system integrators, engineers, scientists, etc. to leverage their C/C++ programming language skills. With Ch, one can develop almost anything from simple scripts to large-scale programs based on the software design of C/C++.

1.2 Who Should Use This Document?

This manual is intended for system administrators and Web masters who install software and configure system setups of Ch. After installation, sample Ch programs and Ch CGI scripts can be tested. This manual is also useful for Ch users who want to gain a further understanding of how Ch is installed and setup. Following instructions in this documentation, experienced C/C++ programmers or computer users can quickly get started using Ch.

1.3 How Is This Document Organized?

Chapter 2 gives the system requirements for installing Ch in both Windows and Unix.

Chapter 3 describes how to install and uninstall Ch in Windows. It includes the installation procedure for the electronic version.

Chapter 4 describes how to configure Ch and get started running Ch programs in Windows.

Chapter 5 describes how to install and uninstall Ch in Unix. It includes the installation procedure for the downloaded edition.

Chapter 6 describes how to configure Ch and get started running Ch programs in Unix.

1.4 Typographical Conventions

The following list defines and illustrates typographical conventions used as visual cues for specific elements of the text throughout this document.

- Interface components are window titles, button and icon names, menu names and selections, and other options that appear on the monitor screen or display. They are presented in boldface. A sequence of pointing and clicking with the mouse is presented by a sequence of boldface words.

  Example: Click **OK**

  Example: The sequence **Start->Programs->Ch6.1->Ch** indicates that you first select **Start**. Then select submenu **Programs** by pointing the mouse on **Programs**, followed by **Ch6.1**. Finally, select **Ch**.

- Keycaps, the labeling that appears on the keys of a keyboard, are enclosed in angle brackets. The label of a keycap is presented in typewriter-like typeface.

  Example: Press `<Enter>`

- Key combination is a series of keys to be pressed simultaneously (unless otherwise indicated) to perform a single function. The label of the keycaps is presented in typewriter-like typeface.

  Example: Press `<Ctrl><Alt><Enter>`

- Commands presented in lowercase boldface are for reference only and are not intended to be typed at that particular point in the discussion.

  Example: "Use the **install** command to install..."

  In contrast, commands presented in the typewriter-like typeface are intended to be typed as part of an instruction.

  Example: "Type `install` to install the software in the current directory."

- Command Syntax lines consist of a command and all its possible parameters. Commands are displayed in lowercase bold; variable parameters (those for which you substitute a value) are displayed in lowercase italics; constant parameters are displayed in lowercase bold. The brackets indicate items that are optional.

  Example: **ls** [**-aAbcCdfFgilLmnopqrRstux1**] [*file* ...]

- Command lines consist of a command and may include one or more of the command's possible parameters. Command lines are presented in the typewriter-like typeface.

  Example: `ls /home/username`

- Screen text is a text that appears on the screen of your display or external monitor. It can be a system message, for example, or it can be a text that you are instructed to type as part of a command (referred to as a command line). Screen text is presented in the typewriter-like typeface.

  Example: The following message appears on your screen

```
usage:  rm [-fiRr] file ...


ls [-aAbcCdfFgilLmnopqrRstux1] [file ... ]
```

- Function prototype consists of return type, function name, and arguments with data type and parameters. Keywords of the Ch language, typedefed names, and function names are presented in boldface. Parameters of the function arguments are presented in italic. The brackets indicate items that are optional.

  Example: **double derivative**(**double** (**func*)(**double**), **double** *x*, ... [**double** **err*, **double** *h*]);

- Source code of programs is presented in the typewriter-like typeface.

  Example: The program **hello.ch** with code

```
int main() {
    printf("Hello, world!\n");
}
```

  will produce the output `Hello, world!` on the screen.

- Variables are symbols for which you substitute a value. They are presented in italics.

  Example: module *n* (where *n* represents the memory module number)

- System Variables and System Filenames are presented in boldface.

  Example: startup file **/home/username/.chrc** or **.chrc** in directory `/home/username` in Unix and **C:**\ >**_chrc** or **_chrc** in directory C:\ > in Windows.

- Identifiers declared in a program are presented in typewriter-like typeface when they are used inside a text.

  Example: variable `var` is declared in the program.

- Directories are presented in typewriter-like typeface when they are used inside a text.

  Example: Ch is installed in the directory `/usr/local/ch` in Unix and `C:/Ch` in Windows.

- Environment Variables are the system level variables. They are presented in boldface.

  Example: Environment variable **PATH** contains the directory /usr/ch.

1.5 Other Relevant Documentations

The core Ch documentation set consists of the following titles. These documentation come with the CD and are installed in CHHOME/docs, where CHHOME is the Ch home directory.

- *The Ch Language Environment — Installation and System Administration Guide*, version 6.1, SoftIntegration, Inc., 2008.

  This document covers system installation and configuration, as well as setup of Ch for Web servers.

- *The Ch Language Environment, — User's Guide*, version 6.1, SoftIntegration, Inc., 2008.

  This document presents language features of Ch for various applications.

- *The Ch Language Environment, — Reference Guide*, version 6.1, SoftIntegration, Inc., 2008.

  This document gives detailed references of functions, classes and commands along with sample source code.

- *The Ch Language Environment, — SDK User's Guide*, version 6.1, SoftIntegration, Inc., 2008.

  This document presents Software Development Kit for interfacing with C/C++ functions in static or dynamical libraries.

- *The Ch Language Environment CGI Toolkit User's Guide*, version 3.5, SoftIntegration, Inc., 2003.

  This document describes Common Gateway Interface in CGI classes with detailed references for each member function of the classes.

Chapter 2

System Requirements

This chapter describes the system requirement in both Windows and Unix for installation.

2.1 System Requirement for Windows 95/98/Me/NT/2000/X/Vista

To install and use Ch for Windows, the following minimum requirements should be met:

- Operating System: Windows 95/98/Me/2000/XP/Vista/Windows NT workgroup or Server 4.0 with SP3 or above
- CPU: with a 486 or higher microprocessor
- Memory: a minimum of 16 Megabytes of RAM
- Disk Space: 30 Mb for Ch Standard Edition, 38 Mb for Ch Professional and Student Editions,

2.2 System Requirement for Unix

For Unix, the supported Operating System is

- Intel Linux 2.4.20-8 or above
- Sparc Solaris 2.6 or above
- HP-UX 10.20 or above
- FreeBSD 5.1 or above
- QNX 6.2.1 or above
-

The hardware requirement for the Intel Linux platform is

- Pentium/90Mhz or above
- A minimum of 16 Megabytes of RAM
- Disk Space Requirement. 25 Mb for Ch Standard Edition, 30 Mb for Ch Professional and Student Editions,

2.3 System Requirement for Mac OS X

To install and use Ch for Mac OS X, the following minimum requirements should be met:

- Operating System: Mac OS X 10.3 or above
- CPU: PowerPC
- Memory: a minimum of 16 Megabytes of RAM
- Disk Space: 25 Mb for Ch Standard Edition, 30 Mb for Ch Professional and Student Editions,

Chapter 3

Install Ch in Windows

Before starting the installation, close all running applications. If you have installed an older version or a beta version of Ch before, uninstall it off the system first. You may want to backup the configuration files in directory **CHHOME/config** before installation. **Note that CHHOME is not the string "CHHOME". Rather, it is the Windows filesystem path under which Ch is installed.** For instance use **C:\Ch** for CHHOME in Windows.

3.1 Install Ch from a Downloaded File

You need to have the system administrator privilege for installing **Ch** into the NT/2000/XP/Vista. A general user account should be OK for Windows 95/98/Me.

1. Download the self-extracting file such as **chstandard-6.1.0.exe** for Ch version 6.1 from the SoftIntegration website.
2. When prompted, choose a directory for installation, for example:

   C:\Ch
3. From Windows Explorer, navigate to the directory; then, double-click the **chstandard-6.1.0.exe** file.
4. To complete installation, just follow the prompted instructions.

3.2 Install Ch from a CD

To start the installation process from a CD:

1. Login to the computer with an Administrator privilege under Windows NT/2000/XP/Vista, or login to the computer in Windows 95/98/Me.
2. Insert the Ch setup CD into the CD-ROM drive.
3. On Windows 95/98/Me and Windows NT/2000/XP/Vista, the setup process starts automatically if AutoPlay for CDs is enabled. Click `Next` to continue.

   If AutoPlay for CDs is not enabled, use Windows Explorer to navigate from the root directory of the CD. Then, double-click the `Setup.exe` file.
4. Read and accept the SoftIntegration license agreement.

5. Enter the product Key (if required)
6. Accept default folder names.
7. Accept the typical installation and press `Next`
8. Follow the instructions of the setup program to install Ch on your computer.
9. Click `Finish` to complete the installation

Note: You are able to quit the installation at any time by pressing the `<Cancel>` button displayed in every dialog box during the installation. You can also move back and forth to review your settings by clicking the `<Back>` and `<Next>` buttons.

3.3 Uninstall Ch in Windows

Stop all the Ch programs.

Click **Control Panel** in **My Computer**. Click **Add/Remove Programs**, select **SoftIntegration Ch 6.1 xxx Edition** for Ch version 6.1, where `xxx` can be `Professional`, `Standard`, `Student`, then Click **Add/Remove ...**. Press **Yes** if you are asked to completely remove Ch and all of its components.

Delete the start up file **_chrc** or **_chsrc** file in **C:\>** or your home directory. For Windows NT/2000/XP/Vista, you need to remove the environment variable settings for Ch. Click **Start**, **Settings**, **Control Panel**, **System**, **Environment**, select the environment variable **PATHEXT**, and delete the added pathext **.ch**.

Chapter 4

System Administration and Getting Started in Windows

This chapter addresses the setup, system administration, and startup of Ch in Windows 95/98/Me/NT/2000/XP/Vista.

4.1 Windows Environment Settings

This section explains how the environment variables are set, and what changes you can make if you want to.

In general, Ch installation will take care of any path or configuration settings, and you can ignore this section if you are not interested in how the environment settings are done in Windows.

Ch will create and set CHHOME in its registry upon installation. CHHOME is the directory where Ch is installed. For Ch Standard and professional edition, the default CHHOME value is C:\Ch.

For Windows 95/98/Me, Ch will copy CHHOME/bin/ch.exe and CHHOME/bin/chs.exe to the windows system directory X:/WINDOWS, where X is the drive where Windows is installed. Also, ch.exe and chs.exe will copy to X:/WINDOWS/ as ch and chs respectively.

For Windows NT/2000/XP/Vista, CHHOME/bin/ch.exe and CHHOME/bin/chs.exe will be copied to windows system directory such as X:/WINNT/SYSTEM32, where X is the drive where Windows is installed. Also, ch.exe and chs.exe will copy to X:/WINNT/SYSTEM32/ as ch and chs respectively.

Upon uninstallation, the above changes will be undone. If you reinstall or upgrade the Ch software again into the same directory as before, you will have to reboot the computer for the changes to take effect.

4.2 Startup

Once you have downloaded and installed the software according to the installation instruction, you can begin to write and run the C code with enhanced functionality. A Ch program typically has the file extension **.ch**. Ch can give you an interactive way of writing and running programs. You can get into either a **regular Ch** or **safe Ch** language environment.

There are four ways to get into the Ch language environment.

1. Click the icon **Ch Standard**, **Ch Professional**, **Ch Student**, on the Desktop screen to get into the regular Ch shell of the corresponding edition, similar to MS-DOS.
2. Click **Start->Programs->SoftIntegration Ch #.# XXX->Ch XXX**, where **#.#** is the version number and **XXX** is one of editions **Standard**, **Professional**, and **Student**.
3. Click **Start**, followed by **Run**, then type `ch.exe`, `chs` or `ch -S`.
4. Go to the MS-DOS prompt, and type `ch`. This turns the MS-DOS shell into an interactive Ch shell.

Note: `chs` and `ch -S` is the same in functionality for the safe Ch. However, it is recommended to use `ch -S` whenever possible for high efficiency.

C code can run directly in Ch shell. C programs with file extension **.c** can also run without compilation.

By running command `ch`, you can get into the regular Ch shell. By running command `ch -S`, you can get into the safe Ch shell. Safe Ch shell disables the pointer and many other functions, such as **system()**, which may jeopardize the security of the system.

4.2.1 Command Line Options

A noninteractive **Ch** shell can execute a command supplied as an argument on its command line. Except the following options, the remaining words from the command line are passed as arguments to the invoked command.

-S Safe Ch shell. Many functions such as **system()** are not available for safe shell.

-a Portable code such as applets. Platform-dependent functions in CHHOME/lib/libopt cannot be used.

-c Read commands from the first filename argument (which must be present and readable). The remaining arguments are passed as arguments to _argv. If the program is a Ch command with function **main**(*int argc, char *argv[]*), arguments will also pass to argv of function **main()**.

-d When ch is started, it first checks if file _chrc exists in the user's home directory. If not, Ch will copy CHHOME/config/_chrc to the user's home directory. When chs is started, it first checks if file _chsrc exists in the user's home directory. If not, Ch will copy CHHOME/config/_chsrc to the user's home directory.

-f Fast start. Read and execute neither **chrc** nor **_chrc** files. If starting a safe Ch shell, it will neither read and execute **chsrc** file nor **_chsrc**.

-g For CGI script debug. It turns the Web browser into a text shell.

-i Reserved for forced interactive shell (ignored).

-n Parse (interpret), but do not execute commands. This option can be used to check Ch shell scripts for syntax errors. The **_warning** flag will be set to level 1. All warning messages will be printed out. Start up files will be parsed only without execution.

-r Redirect stderr stream to stdout. This option is especially useful for debugging programs running in Windows operating systems. For example, command `ch -r chcmd >junkfile` will send error messages from stderr stream in program **chcmd** to file **junkfile**. For Unix, you can still use 2>&1 to redirect the stderr.

-v Print out Ch edition and version number in the stdout stream.

-w The _warning flag will be set to the highest level for both parsing and execution of the program. All warning messages will be printed out.

4.2.2 Home Directory

At the initialization of Ch, the system variable **_home** for the user's home directory will get its value according to the following algorithm:

- If environment variable **HOME** has been setup, the value of $HOME will be used.
- If environment variables **HOMEDRIVE** and **HOMEPATH** have been setup, these value will be used to form the user's home directory.
- If the root of the windows directory, such as **C:**\, is writable to the user, it becomes the home directory.
- If directory **X:**\ is writable to the user, where X is the drive to which Ch is installed, **X:**\ will be the home directory.
- If none of the above is valid, initialization of Ch will fail.

The environment variable HOME can be setup in file **C:****autoexec.bat** in Windows 95/98. Add the following line into your **autoexec.bat** file:

```
SET HOME=C:\
```

In Windows NT/2000/XP/Vista, it can be setup by clicking `Control Panel`, `System`, then `Environment`. If you do not set environmental variable `HOME` manually, Ch will set `HOME` to the value of environmental variables `%HOMEDRIVER%%HOMEPATH%`. If the value for `%HOMEDRIVER%%HOMEPATH%` is empty, the `HOME` will be set to your current working directory.

4.2.3 Startup Files

Assume the environment variable **CHHOME** is the top diretory where Ch is installed. Its default value is **C:****Ch** in Windows. The following startup files are executed when the Ch language environment is invoked.

CHHOME/config/chrc	Invoked automatically upon starting Ch.
CHHOME/config/chsrc	Invoked automatically upon starting safe Ch.
_home/_chrc	User's local file invoked by chrc.
_home/_chsrc	User's local file invoked by chsrc.

There are four built-in system variables that can be used to set search paths, **_path**, **_lpath**, **_fpath** and **_ipath**. They can be setup in the configuration file **_chrc**.

- **_path**: A built-in system variable in Ch used to search the binary commands and executable script files.
- **_lpath**: The path for dynamically loaded libraries.

- **_fpath**: The function path. The default path for safe Ch shell is **CHHOME/lib/libc** and **CHHOME/lib/libch**; while the default function path for regular Ch shell is **CHHOME/lib/libc, CHHOME/lib/libch,** and **CHHOME/lib/libopt;**

  Functions not located in the above directories cannot be used in startup files **_chrc** and **_chsrc**. But, generic functions can be used in these startup files.

- **_ipath**: the path for included header files.

There is also an environment variable **PATH** which has the same value as system variable **_path** in Ch. Some programs, such as `command.com` and `cmd.exe`, use it to search for other programs.

4.2.4 Startup Ch

For Windows 95/98/NT/2000/XP/Vista, type the command `ch` from MS-DOS Shell or click the ch icon to get into the regular Ch. The startup files are executed according to the following pseudo algorithm.

```
execute CHHOME/config/chrc
if  _home\_chrc exists
then
     call _home\_chrc
endif
```

If you start Ch as a safe shell, run the command `ch -S` or `chs`.

```
execute CHHOME/config/chsrc
if  _home\_chsrc exists
then
     call _home\_chsrc
endif
```

If you start Ch with argument `-f`, similar to the csh, for the regular Ch, they will bypass CHHOME/chrc and _chrc. For safe Ch shell, they will bypass CHHOME/chsrc and _chsrc.

By default, there is no startup file _chrc in a user's home directory. The system administrator may add a startup file in a user's home directory. However, the user can execute Ch with option -d as follows

```
ch -d
```

to copy a sample startup file from directory CHHOME/config/ to the user's home directory if there is no startup file in the home directory yet.

4.3 Testing Setup and Demos

After Ch is installed, the following commands can be used to test if the setup is correct or not. These commands also demonstrate some Ch features.

- The commands given below will give you some limited help.

```
ch -d
help
```

- The commands given below will give you a quick demo of Ch as a C interpreter.

```
ch
int i = 90
i
i*i+2

int *p1, **p2
p1 = &i
p2 = &p1
i=5
*p1
**p2 = 10
*p1
i
printf("i= %d\n", i)
sin(3.14/2.0)
exit
```

- Similar to C-shell, Ch is also a command shell.

```
ch
pwd
alias cool "echo Ch is cool!"
alias cool
cool
unalias cool
cool
alias("mydir", "cd /bin");
mydir
pwd
alias
```

- Programs and commands are integrated in Ch.

```
dir
dir /b
string_t  s="/b"
dir $s
s = ‘dir /b‘
s
s = ‘‘dir /b‘‘
s
printf("s = %s\n", s)
```

- A quite useful program is **which**. This program, written in Ch, can be used to find commands, function files, header files, and environment variables.

```
which pwd
which dir
which int sin
which unknown
which which
which -a which
which stdio PATH
which -a stdio PATH
```

- IO stream similar to C++

```
int k
cout <<"Type in a number for k"
cin >> k
k
cout << k
```

- Try complex numbers by typing

```
complex z = complex(3, 4)
z+z;
2*z;
sin(z);
abs(z);
```

- Array bounds are checked to prevent memory corruption

```
double a[3] = {1, 2, 3}
a[0]
a[1]
a[2]
a[-1]
a[3]
```

- The range of array subscripts can be adjusted

```
double b[1:3] = {1, 2, 3}
b[0]
b[1]
b[2]
b[3]
b[4]
```

- Try class/struct

```
struct cube{int i, j; float f;} c1 = {1,2,3}
c1
sizeof(c1)
```

```
struct cube c2;
c2 = c1
c2
cube c3
c3
c3.i = 10
c3
2*c3.i
```

- Try union

```
union tag {int i; double d;} m
m.i = 10
m
m.d = 10
m
sizeof(m)
```

- To test Ch applet and safe shell, open the following file **CHHOME/demos/chs/c1.ch** from a Web browser such as Internet Explorer or Netscape. For example, if the environment variable **CHHOME** is set to C:/Ch in Windows, after clicking the **Open** menu on your Web browser, type

  ```
  C:/Ch/demos/chs/c1.chs
  ```

  Then, the output of program **c1.chs** "Hello, world!" and today's date should be displayed. Note that the Web browser might prompt you with two options, **Save** and **Open** the file. Click **Open** in this case, the program will be executed.

Chapter 5

Install Ch in Unix

If you have installed an older version or a beta version before, uninstall that version off the system first. You may want to backup some configuration files in directory **CHHOME/config** that you have modified in the older version. **Note that CHHOME is not the string "CHHOME". Rather, it is the Unix filesystem path under which Ch is installed**. Under Unix, the default directory for installing Ch version 6.1 is **/usr/local/ch6.1**, and the symbolic /usr/local/ch, /opt/ch, or /usr/ch will be created, and CHHOME will be set to one of the directories /usr/local/ch, /opt/ch, and /usr/ch.

5.1 Install Ch from a Downloaded File

1. Download the compressed file from the SoftIntegration website.
2. When prompted, choose a temporary directory for the **ChEdition-Version.OSversion.platform.tar.gz** file, for example:

   /tmp

   where Platform and OSversion will be substituted by a real platform and OS version, for example, **chstandard-6.1.0.solaris2.6.sparc.tar.gz** for Ch version 6.1 for Sun Sparc station with Solaris 2.6 or higher. The recommended directory to install Ch is /usr/local/ch6.1 or HOME/ch.
3. For Ch version 6.1, run the following command from the temporary directory, and follow the prompted instructions.

```
gzip -cd  chstandard-6.1.0.solaris2.6.sparc.tar.gz | tar xvof -
cd chstandard-6.1.0.solaris2.6.sparc
sh ./install.sh
```

5.2 Install Ch from a CD

If you have the CD with you, install using the following steps.

1. Login as root
2. Insert the Ch setup CD into the CD-ROM drive. Depending on how your operating system is configured, your CD drive may be mounted automatically. If the CD drive is not mounted, you must mount it before continuing.

3. Go to your CD-ROM directory where the CD-ROM is mounted.
4. Run the following command.

```
sh ./install.sh
```

5.3 Uninstall Ch in Unix

Take the following steps:

- Remove all Ch and its components from the CHHOME directory where you installed Ch.
- Check /etc/shells to see if login shells /bin/ch and /bin/chs are deleted.
- Check if startup files $HOME/.chrc and $HOME/.chsrc are deleted.
- Check if the symbolic links /usr/ch, /usr/local/ch, /opt/ch /bin/ch, and /bin/chs are deleted

Chapter 6

Install Ch in Mac OS X

You need to have the system privilege as a superuser to install Ch on Mac OS X. If you have installed an older version before, uninstall that version off the system first. You may want to backup some configuration files in directory **CHHOME/config** that you have modified in the older version. **Note that CHHOME is not the string "CHHOME". Rather, it is the Mac OS X filesystem path under which Ch is installed**. In the following description, we assume Ch version 6.1 will be installed or uninstalled. For a different version, change the numerical version number 6.1 to a different number. Under Mac OS X, the default directory for installing Ch version 6.1 is **/usr/local/ch6.1**. For different versions, the numerical number following **/usr/local/ch** will be different. The symbolic /usr/local/ch will be created, and CHHOME is /usr/local/ch.

6.1 Install Ch from a Downloaded File

1. Download the compressed file from the SoftIntegration website.
2. Go to Unitlies, then click Terminal.
3. Change to the directory where your downlaoded file such as chstandard-6.1.0.macosx10.ppc.tar.gz is located. For example, if it is located on your desktop, by command

   ```
   cd /Users/your_account_name/Desktop
   ```

4. You can untar and decompress the downloaded file with the command below.

   ```
   gzip -cd  chstandard-6.1.0.macosx10.ppc.tar.gz | tar xvf -
   cd chstandard-6.1.0.macosx10.ppc
   sudo sh ./install.sh
   ```

6.2 Install Ch from a CD

If you have the CD with you, install using the following steps.

1. Insert the Ch setup CD into the CD-ROM drive. Depending on how your operating system is configured, your CD drive may be mounted automatic ally on your Desktop.
2. Run the following command.

```
sudo sh ./install.sh
```

then you can follow the instructions to install.

6.3 Setup for Plotting Using AquaTerm

Plots in Ch Professional Edition can be displayed using either X11 or AquaTerm.

Plots in Ch are displayed using X11 by default. Installation instructions for X11 can be found by searching for "X11 install" on Apple Computer's web site http://www.apple.com.

AquaTerm is an open source application for Mac OS X that provides a GUI interface for plotting programs. To use AquaTerm for displaying plots in Ch, follow the instructions below to set it up.

1. Downloaded AquaTerm from the internet at http://aquaterm.sourceforge.net and install it.
2. Create a system startup file **.chrc** in your home directory by command

   ```
   ch -d
   ```

3. Add the statement

   ```
   putenv("GNUTERM=aqua");
   ```

   inside the startup file **.chrc** in your home directory. A line may already exist and just need to be uncommented. If this is the case, just uncomment the line by getting rid of the "//" at the beginning of the line.

6.4 Uninstall Ch in Mac OS X

You will have to be the root user for uninstalling Ch.

- Remove /usr/local/ch6.1 by command

  ```
  sudo rm -rf /usr/local/ch6.1
  ```

- Remove the symbolic links /usr/local/ch, /bin/ch, and /bin/chs by commands

  ```
  sudo rm -f /usr/local/ch
  sudo rm -f /bin/ch
  sudo rm -f /bin/chs
  ```

- Check /etc/shells to see if login shells /bin/ch and /bin/chs are deleted
- Check if startup files $HOME/.chrc and $HOME/.chsrc are deleted

Chapter 7

System Administration and Getting Started in Unix and Mac OS X

This chapter addresses the setup, system administration, and startup of Ch in Unix.

7.1 Unix and Mac OS X Configuration

Upon installation, the Ch installer will create two symbolic links **/bin/ch** and **/bin/chs** which point to the installed binary files. Also, they will append /bin/ch and /bin/chs to the file **/etc/shells** for ftp to be able take ch as a login shell if you selected **yes** upon installation for the prompted questions.

Ch can run as a login shell similar to the C-Shell and Bash under Unix. You can become a root and modify the password file **/etc/passwd** to make ch as a default login shell for any user account. For example,

```
tempuser:X:500:500:temp user:/home/tempuser:/bin/csh
```

changed to either regular Ch shell :

```
tempuser:X:500:500:temp user:/home/tempuser:/bin/ch
```

or safe shell:

```
tempuser:X:500:500:temp user:/home/tempuser:/bin/chs
```

7.2 Internet Computing

Ch is denoted by a specific file extension. **.ch** as the default Ch file extension, **.chs** is the safe Ch file extension. Both the Web browser and server can be configured to take advantage of internet computing.

7.2.1 Web Browser Configuration

1. Copy the file **CHHOME/config/.mime.types** to your home directory or append the following to your existing file ~**/.mime.types** in the user's home directory

   ```
   # handle CH language environment
   application/x-chs            chs
   ```

2. Then, copy the file **CHHOME/config/.mailcap** to your home directory or append the following to your existing file ~**/.mailcap** in the user's home directory.

```
#handle CH language environment
application/x-chs; ch -S %s
```

When file ~**/.mailcap** in user's home directory is changed, the Web browser needs to be restarted to make it effective.

7.3 Startup

Once you have downloaded and installed the software according to the installation instruction, you can get into either the **regular Ch** or **safe Ch** language environment.

The **ch** and **chs** shells are similar to csh, and you will find it much easier if you are already familiar with csh or tcsh.

You can type `ch` to get into the regular Ch, `ch -S` or `chs` to the safe Ch from your Unix shell. (**Note**: **chs** is equivalent to **ch -S**, However, it is recommended to use `ch -S` whenever possible for efficiency.)

Note: HOME is the user's home directory while CHHOME is the directory where Ch is installed. By default, CHHOME is /usr/ch or /usr/local/ch.

Running command `ch` can take you into regular Ch shell while `ch -S` can get you into the safe Ch shell. Safe Ch shell disables the pointer and many other functions, such as **system()**, which may jeopardize the security of the system,

When regular Ch is started,

```
execute CHHOME/config/chrc
if  HOME/.chrc exists
then
     call HOME/.chrc
endif
```

When safe Ch is started,

```
execute CHHOME/config/chsrc
if  HOME/.chsrc exists
then
     call $HOME/.chsrc
endif
```

Ch can also be used as a login shell in Unix and plays the similar role as a Unix Shell such as sh, bash, csh and tcsh. You can change login shell to either **/bin/ch** or **/bin/chs** in **/etc/passwd**. If the shell is invoked with a name that starts with '-', as when started by login(1), the shell runs as a login shell. In this case, the regular Ch login shell will start as the follows:

```
execute CHHOME/config/chrc
if  HOME/.chrc exists
then
     call HOME/.chrc
endif
execute CHHOME/config/chlogin
if HOME/.chlogin exits
then
     call HOME/.chlogin
endif
```

For the safe Ch login shell, it starts:

```
execute CHHOME/config/chsrc
if  HOME/.chsrc exists
then
     call HOME/.chsrc
endif
execute CHHOME/config/chslogin
if HOME/.chslogin exits
then
     call HOME/.chslogin
endif
```

Typically, the file **.chlogin** and **.chslogin** contain commands to specify the terminal type and environment.

As a login shell terminates, the regular Ch shell will execute the commands from **$HOME/.chlogout**, and the safe Ch shell will execute the commands from **$HOME/.chslogout**.

By default, there is no startup file .chrc in a user's home directory. The system administrator may add a startup file in a user's home directory. However, the user can execute Ch with option -d as follows

```
ch -d
```

to copy a sample startup file from directory CHHOME/config/ to the user's home directory if there is no startup file in the home directory yet.

7.3.1 Command Line Options

A noninteractive **Ch** shell can execute a command supplied as an argument on its command line. Except for the following command line options, the remaining words from the command line are passed as arguments to the invoked command.

- S Safe Ch shell. Many functions such as **system()** are not available for safe shell.

- a Portable code such as applets. Platform-dependent functions in CHHOME/lib/libopt cannot be used.

- c Read commands from the first filename argument (which must be present and readable). Remaining arguments are passed as arguments to _argv. If the program is a Ch command with function **main**(*int argc, char *argv[]*), arguments will also pass to argv of function **main()**.

- d When ch is started, it first checks if file .chrc exists in the user's home directory. If not, Ch will copy CHHOME/config/.chrc to the user's home directory. When chs is started, it first checks if file .chsrc exists in the user's home directory. If not, Ch will copy CHHOME/config/.chsrc to user's home directory. In Windows, startup files _chrc and _chsrc instead of .chrc and .chsrc, will be used for regular Ch and safe Ch, respectively.

- f Fast start. Read neither the **chrc** and **.chrc** files, nor the **chlogin** and **.chlogin** files (if a login shell) upon startup.

- g For CGI script debug. It turns the Web browser into a text shell.

- i Reserved for forced interactive shell (ignored).

- n Parse (interpret), but do not execute commands. This option can be used to check Ch shell scripts for syntax errors. **_warning** flag will be set to level 1. All warning messages will be printed out. Start up files will be parsed only without execution.

- r Redirect stderr stream to stdout. This option is especially useful for debugging programs running in Windows operating systems. For example, command **ch -r chcmd** > **junkfile** will send error messages from stderr stream in program **chcmd** to file **junkfile**. For Unix, you can still use 2>&1 to redirect the stderr.

- v Print out Ch edition and version number in the stdout stream.

- w The _warning flag will be set to the highest level for both parsing and execution of the program. All warning messages will be printed out.

7.3.2 Startup Files

Assume the environment variable **CHHOME** is the top diretory where Ch is installed. Its value is **/usr/ch** in Unix. The following startup files are executed when the Ch language environment is invoked.

CHHOME/config/chrc	Invoked automatically upon starting Ch.
CHHOME/config/chlogin	Invoked by Ch automatically after chrc when Ch is used as login shell.
CHHOME/config/chlogout	Invoked by Ch automatically while Ch logout.
CHHOME/config/chsrc	Invoked automatically upon starting safe Ch.
CHHOME/config/chslogin	Invoked by safe Ch automatically after chsrc when safe Ch is used as a login shell.
CHHOME/config/chslogout	Invoked by safe Ch automatically when safe Ch logout.
CHHOME/config/.mime.types	Used by Web browser
CHHOME/config/.mailcap	Used by Web browser.
~**/.chrc**	User's local file invoked by chrc.
~**/.chlogin**	User's local file invoked by chlogin.
~**/.chlogout**	User's local file invoked by regular Ch login shells at logout.
~**/.chsrc**	User's local file invoked by chsrc.
~**/.chslogin**	User's local file invoked by chslogin.
~**/.chslogout**	User's local file invoked by safe Ch login shells at logout.

There are four built-in system variables can be used to set search paths, **_path**, **_lpath**, **_fpath** and **_ipath**. They can be setup in the configuration file **.chrc**.

- **_path**: A built-in system variable in Ch used to search the binary command s and executable script files.
- **_lpath**: The path for dynamically loaded libraries.
- **_fpath**: The function path, the default path for safe Ch shell is **/usr/ch/lib/libc** and **/usr/ch/lib/libch**. while the default path for regular Ch shell is **/usr/ch/lib/libc, /usr/ch/lib/libch** and **/usr/ch/lib/libopt**.

  Functions not located in the above directories cannot be used in startup files **.chrc** and **.chsrc** But, generic functions can be used in these startup files.
- **_ipath** the path for included header files.

There is also an environment variable **PATH** which has the same value as system variable **_path** in Ch . Some programs, such as sh, use it to search for other programs.

7.4 Testing Setup and Demos

After Ch is installed, the following commands can be used to test if the setup is correct or not. These commands also demonstrate some of Ch features.

- The commands given below will give you some limited help.

```
ch -d
help
```

- The commands given below will give you a quick demo of Ch as a C interpreter.

```
ch
int i = 90
i
i*i+2

int *p1, **p2
p1 = &i
p2 = &p1
i=5
*p1
**p2 = 10
*p1
i
printf("i= %d\n", i)
sin(3.14/2.0)
exit
```

- Similar to C-shell, Ch is also a command shell.

```
ch
pwd
alias cool "echo Ch is cool!"
alias cool
cool
unalias cool
cool
alias("mydir", "cd /bin");
mydir
pwd
alias
```

- Programs and commands are integrated in Ch.

```
ls
ls -1
int j =-1
ls $j
string_t s
s = ‘ls‘
s
s = ‘‘ls‘‘
s
printf("s = %s\n", s)
```

- A quite useful program is **which**. This program, written in Ch, can be used to find commands, function files, header files, and environment variables.

```
which pwd
which ls
which int sin
which unknown
which which
which -a which
which stdio PATH
which -a stdio PATH
```

- IO stream similar to C++

```
int k
cout <<"Type in a number for k"
cin >> k
k
cout << k
```

- Try complex numbers by typing

```
complex z = complex(3, 4)
```

```
z+z;
2*z;
sin(z);
abs(z);
```

- Array bounds are checked to prevent memory corruption

```
double a[3] = {1, 2, 3}
a[0]
a[1]
a[2]
a[-1]
a[3]
```

- The range of array subscripts can be adjusted

```
double b[1:3] = {1, 2, 3}
b[0]
b[1]
b[2]
b[3]
b[4]
```

- Try class/struct

```
struct cube{int i, j; float f;} c1 = {1,2,3}
c1
struct cube c2;
c2 = c1
c2
cube c3
c3
c3.i = 10
c3
2*c3.i
```

- Try union

```
union tag {int i; double d;} m
m.i = 10
m
m.d = 10
m
sizeof(m)
```

- To test a Ch applet and safe shell, open the following file **CHHOME/demos/chs/c1.chs** from a Web browser such as Internet Explorer or Netscape. For example, if the environment variable **CHHOME** is set to /user/ch, after clicking the **Open** menu on your Web browser, type

```
/usr/ch/demos/chs/c1.chs
```

Then, the output of program **c1.chs** "Hello, world!" and today's date will be displayed. If you have not followed the setup steps outlined in section 7.2 on page 20, program **c1.chs** will be displayed instead of being executed.

Index