Installation and Administrator's Guide

Together® ControlCenter™ Version 6.0

TABLE OF CONTENTS

Configuring Together for Multiple Users 23

Configuring Options 29

Using the Together License Server 38

PREFACE

The *Installation and Administrator's Guide* contains information for installing and configuring Together and information for setting-up and maintaining the Together floating license server.

Audience

The information in this guide is intended primarily for the site administrator responsible for Together, in addition to any user who needs to install Together. The documentation set for Together assumes that you have experience with the following:

- The basics of the Unified Modeling Language™ (UML™).
- Your organization's process for designing and modeling object-oriented systems or components.
- Your target programming language(s).
- Fundamentals of development technologies such as Enterprise JavaBeans™ (if you will use Together to develop and deploy an EJB™, JSP™, etc.)
- Your external tools and systems such as IDE's, version control systems, database management systems (DBMS), application servers, etc.
- How to install a Java™ VM and launch a Java application with it

Together Documentation Set

The documentation set for Together consists of the items listed in Table 1.

> **NOTE:** To view the *Installation and Administrator's Guide* and the *User Guide* which are available as PDFs, you need Acrobat Reader. To download a free version of Acrobat Reader, visit the Adobe website at http://www.adobe.com/products/acrobat/readermain.html.

Table 1 Together documentation

Item	Description	Location
Together 6.0 Readme	Late-breaking information including: • Licensing notes • System requirements • Installing and starting Together • Known problems • Significant fixes	Your Together installation under `$TGH$/readme.html`

Table 1 Together documentation (continued)

Item	Description	Location
Installation and Administrator's Guide	Information most relevant to the Together administrator, including: • System requirements • Installing and configuring Together • Setting-up and using Together's floating license server	PDF located as follows: • Your Together installation under `$TGH$/doc/installGuide.pdf` • http://www.togethersoft.com/downloads
User Guide	Information most relevant to the user, including: • Introduction to Together • Setting personal preferences and options • Detailed instructions for using Together features • Advanced customizations • Reference information including commands, macros, and keyboard shortcuts	PDF located as follows: • Your Together installation under `$TGH$/doc/userGuide.pdf` • http://www.togethersoft.com/downloads
Online Help	Information most relevant to the user, including: • Main window components • Brief instructions for using Together features with references to the *User Guide* for more details • Dialog help	Together's Help menu
Documentation for integrations with third-party products	Instructions for using Together integrations and plug-ins.	http://www.togethersoft.com/developers/integrations
API documentation	Technical reference for the Together API generated by Javadoc; Use in conjunction with API information in the *User Guide*.	• Together's Help menu • Your Together installation under `$TGH$/doc/api/index.html`
Practical guides	A collection of interactive tutorials that provide an introduction to Together for the new user.	http://www.togethersoft.com/services/practical_guides/

Finding out more

If you are just getting started with UML, object technology, or distributed application development, the following books and resources can help get you started.

Third-Party Books

UML Distilled: Applying the Standard Object Modeling Language by Martin Fowler. Addison-Wesley, 1997. ISBN: 0201325632

The Unified Modeling Language User Guide by Booch, Rumbaugh, and Jacobson. Addison-Wesley, 1998. ISBN 0-201-57168-4

Java Modeling in Color with UML: Enterprise Components and Process by Peter Coad, Jeff De Luca, and Eric Lefebvre. Prentice Hall, June 1999. ISBN: 013011510X

Java Design: Building Better Apps & Applets (2nd Ed.) by Peter Coad and Mark Mayfield. Prentice Hall, 1997. ISBN 0-13-271149-4 (pbk).

Java Enterprise in a Nutshell by David Flanagan, Jim Farley, William Crawford, and Kris Magnusson. O'Reilly & Associates, 1999. ISBN 1-56592-483-5

On the Web

Object Management Group: http://uml.shl.com/

Cetus Links site: http://www.cetus-links.org

Dev-x Developer Exchange: http://www.devx.com/

Workshops, Mentoring, and Consulting

If your organization needs help making a smooth transition into the world of object and component software technologies, TogetherSoft Corporation offers a wide range of workshops, seminars, mentoring, and consulting services to teams worldwide. For more information, visit the Services area at the TogetherSoft website at www.togethersoft.com/services/.

Getting Help

The TogetherSoft web site offers a comprehensive range of company, product, and technical information.

TogetherSoft corporate site

The TogetherSoft corporate site is located at http://www.togethersoft.com and provides company and product information such as:

- What features come with what Together products (www.togethersoft.com/together)
- Together installation, startup & readme notes (www.togethersoft.com/readme)
- Worldwide Together technical support centers (www.togethersoft.com/support)
- Downloads: products, docs, JVMs, etc. (www.togethersoft.com/downloads)
- Integrations and plug-ins (www.togethersoft.com/developers/integrations)
- Newsletters: info, subscribe, unsubscribe (www.togethersoft.com/subscribe)

Together community site

The Together community site is located at http://www.togethercommunity.com and provides technical information, discussions, open-source code, and more including:

- FAQs
- Discussion forums and e-mail lists.
- Technical articles and presentations on modeling, Java, using Together, and more.
- Open source and binaries for modules and patterns.

CHAPTER

1

INSTALLING AND STARTING TOGETHER

This chapter provides the system requirement, installation, and start-up instructions for the following operating systems (OS):

> **NOTE:** All Together products have a built-in licensing mechanism for node-locked licenses (license files have the extension `.tg`). Together can obtain a node-locked license automatically from the TogetherSoft license server. Beginning with version 6.0, floating licenses are managed by the Together floating license server. Complete licensing documentation is provided in Chapter 4, "Using the Together License Server".

Windows systems

This section contains system requirements, installation, and start-up information for using Together on systems running Microsoft® Windows XP Professional, Windows NT, Windows 2000, Windows ME, Windows 98, or Windows 95.

System requirements

Table 2 lists the system requirements for using Together with Windows.

Table 2 System Requirements for Windows

	RESOURCE	REQUIREMENTS	NOTES
Hardware	CPU	Pentium-II, 350 MHz or faster	Recommended: Pentium-III, 500 MHz or faster
	Memory	256 MB minimum	512 MB or more is recommended when generating project-wide doc for large-scale projects
	Disk Space	NTFS: 180 MB FAT: 180-360 MB (depending on the cluster size)	
	Video	SVGA, high color mode, 1024x768	Recommended resolution: 1280x1024 or more
	CD Drive	Optional	Necessary only if installing from CD
Software	OS	Windows XP Professional, Windows NT 4.0, Windows 2000, Windows ME, Windows 98, Windows 95	**Service Pack 4** or above is required for Windows NT 4.0. Under Windows 95/98/ME, it is recommended to increase the Environment space to 4096.
	Java SDK	Sun® Java2™ SDK version 1.3	
	License Manager	Together floating license client (built-in) and valid node-locked or floating license	Running license server on remote license host with valid license is required for floating licenses.
	Web Browser	MS Internet Explorer 5.0 (or later), or Netscape Navigator 4.6 (or later)	Needed for viewing API documentation
Other	An internet connection is useful for obtaining software updates, node-locked licenses, and support, and for accessing TogetherSoft on-line resources.		

JVM information for Windows

When running under Windows (XP, NT, 2000, ME, 98, and 95), Together requires Sun Java2 SDK (formerly JDK) to support some development features. J2SDK 1.3 is automatically installed on Windows systems and Together runs under its JRE. This is the recommended Java Virtual Machine (JVM) for Together running on Windows.

Installing on Windows

For Windows installation, an executable installation program guides you through the process, and the Together.exe launcher is written to the *bin* directory of the installation.

Windows installation archives

There are currently 2 installer program options for Windows:

- `Together_nnnn.exe`

  Together program and documentation, plus JDK/JRE
- `Together_nnnn_novm.exe`

  Together program and documentation without JDK/JRE

If J2SDK 1.3 is already installed on your system, you can choose to run the partial installer program that does not include the JDK (`Together_nnnn_novm.exe`). During Together installation, you will be asked to specify the Java VM on your system that will be used by Together.

If you do not have J2SDK 1.3 on your system, you need to choose the full installation (`Together_nnnn.exe`).

How to install

The installer comes as a Windows executable file. The file name for the full installation is `together_nnnn.exe` (where *nnnn* is a build number). The file name for the partial installation appends a descriptor to indicate the nature of the installation (such as "novm").

To install Together under Windows:

1. If installing to NT or 2000, make sure you are logged in with Administrator rights.
2. Run the installer program from Windows Explorer or Control Panel's Add-Remove Programs.
3. Follow the on-screen instructions to complete the installation.

Starting Together on Windows

If you have a floating license, you will need to install the Together floating license server on a remote network host accessible to the computer running Together. For complete instructions, see Chapter 4, “Using the Together License Server”. Once license management has been configured, you can start Together as described below.

The installation writes the following runtime files to `$TOGETHER_HOME$/bin`:

- **Together.exe**

  Native Windows executable. It checks the system memory size, sets any necessary variables, and then searches for a compatible Java virtual machine. The search begins with the Sun JDK installed with Together, continues in the `%JDK%` folder specified in Together.bat, and finally proceeds to the system registry if a JDK has not been found. The launcher runs Together under the first compatible JDK it finds. If it cannot find one, it posts an error message and stops.

- **TogetherCon.exe**

  Native console version of the same program, installed under Windows only. It can be helpful to use this version if you want to redirect console input/output to a file. Run `Together.exe/?` or `TogetherCon.exe/?` to see the launchers' parameters.

- **Together.bat**

  Windows batch file. It sets necessary environment variables to default values and launches Together using the Sun JDK installed with Together. You can edit the file to control all the launch parameters if you wish.

To launch Together under Windows:

1. For floating licenses, make sure that the Together floating license server (either local or remote) is properly configured and running. Evaluation licenses do not require the server.
2. Run **TG_HOME\bin\Together.exe**. The executable looks for the Java runtime automatically installed with Together on Windows computers. It also looks for the floating license client (if necessary) and starts it automatically. Alternatively, you can use the Start menu shortcut, which points to this launcher.

Alternatively, if you want to customize memory settings, runtime environment path, and so on, you can edit the file `$TG_HOME$\bin\Together.bat` and then use that file to launch Together in Step 2. If you want to see console output, use `TogetherCon.exe`. Both launchers automatically start the floating license client if it is necessary for your license.

> **NOTE:** Together.exe and TogetherCon.exe use Together.bat as the configuration file. They set all environment variables from this file, except the Java Machine call command.

Solaris systems

This section contains system requirements, installation, and start-up information for getting started using Together on systems running the Solaris operating system.

System requirements

Solaris runs on both Intel and SPARC processors. The system requirements for SPARC workstations are listed in this section. Solaris on Intel is not currently tested or supported – see the section “Other UNIX systems” on page 21. Table 3 lists the system requirements for using Together with Solaris.

Table 3 System Requirements for Solaris

	RESOURCE	REQUIREMENTS	NOTES
Hardware	CPU	SPARC Ultra-5 333 MHz or faster	
	Memory	256 MB minimum	512 MB or more is recommended when generating project-wide doc for large-scale projects.
	Disk Space	180 MB	
	Video	X Window System, Protocol version 11, Revision 6, minimum 1024x768 screen resolution	Recommended resolution: 1280x1024 or more
	CD Drive	Optional	Necessary only if installing from CD
Software	OS	SunOS Release 5.7 (Solaris 7) or later	Recommended SunOS Release 5.8 (Solaris 8)
	Java SDK	Sun® Java2™ Standard Edition (SDK 1.3.1_02) plus: OS patches for Solaris (required to run SDK 1.3.1_02)	Together runs under this JRE. It is installed automatically during Together installation. Patches must be installed separately.
	License Manager	Together floating license client (built-in) and valid node-locked or floating license	Running license server on remote license host with valid license is required for floating licenses.
	Web Browser	MS Internet Explorer 5.0 (or later), or Netscape Navigator 4.6 (or later)	Needed for viewing API documentation
Other	An internet connection is useful for obtaining software updates, node-locked licenses, and support, and for accessing TogetherSoft on-line resources.		

JVM information for Solaris

When running under Solaris, Together requires Sun Java2 SDK (formerly JDK) to support some development features. This is the recommended Java Virtual Machine (JVM) for Together running on Solaris. J2SDK 1.3.1_02 is included in the Together installation for Solaris, and Together runs under its JRE.

> **NOTE:** The bundled JVM will not work unless all the patches for Solaris are installed. *The necessary patches are not installed automatically.* Before installing Together, make sure that you have installed all the OS patches corresponding to

your Solaris version. You can obtain the patches and documentation at: http://java.sun.com/j2se/1.3/install-solaris-patches.html.

Installing on Solaris

Together installation is carried out using the InstallAnywhere utility. The name of the installation file is `Together_nnnn_solaris.bin`, where *nnnn* is a build number.

To install Together on Solaris:

1. Set executable permission rights for the installation file:

   ```
   chmod +x Together_nnnn_solaris.bin
   ```

2. Run the installation file:

   ```
   ./Together_nnnn_solaris.bin
   ```

3. Follow the on-screen instructions to complete the installation.

   NOTE: For Solaris, it is strongly recommended to install the JVM that is included in the Together installer (choose the option “Install a JVM specifically for this application” during the installation setup process).

Starting Together on Solaris

If you have a floating license, you will need to install the Together floating license server on a remote network host accessible to the computer running Together. For complete instructions, see Chapter 4, “Using the Together License Server”. Once license management has been configured, you can start Together as described below.

Launcher

The launcher file `$TG_HOME$/bin/Together.sh` is written to disk during installation. It sets appropriate environment variables including JAVA_HOME (required), and launches Together. You can edit the file to point to a different Java runtime location if you wish. You may have to edit the startup scripts in this file depending on the location to which you install it.

To start Together on Solaris:

1. For floating licenses, make sure that the Together floating license server is properly configured and running.
2. Run **TG_HOME/bin/Together.sh**. If necessary for your license, this launcher automatically starts the floating license client (installed with Together).

LINUX systems

This section contains system requirements, installation, and start-up information for getting started using Together on systems running the LINUX operating system.

System requirements

These requirements apply to LINUX systems and, except for OS and JVM/JDK, to Solaris and other UNIX flavors running on the Intel hardware platform. Table 4 lists the system requirements for using Together with LINUX.

Table 4 System Requirements for Linux

	RESOURCE	REQUIREMENTS	NOTES
Hardware	CPU	Pentium-II, 350 MHz or faster	Recommended: Pentium-III, 500 MHz or faster
	Memory	256 MB minimum	512 MB or more is recommended when generating project-wide documentation on large-scale projects.
	Disk Space	180 MB	200 MB or more swap space recommended
	Video	X Window System, Protocol version 11, Revision 6, minimum 1024x768 screen resolution	Recommended screen resolution: 1280x1024 or more
	CD Drive	Optional	Necessary only if installing from CD
Software	OS	RedHat 6.2 or later[1], or SuSe 7.2[2]	OpenMotif-based Window Manager (such as CDE, MWM) is recommended.
	Java SDK	Sun® Java2™ SDK, Standard Edition, version 1.3 for Linux	Together runs under this JRE. It is installed automatically during Together installation.
	License Manager	Together floating license client (built-in) and valid node-locked or floating license	Running license server on remote license host with valid license is required for floating licenses.
	Web Browser	Netscape Navigator 4.6 (or later)	Needed for viewing API documentation
Other	An internet connection is useful for obtaining software updates, node-locked licenses, and support, and for accessing TogetherSoft on-line resources.		

1.For RedHat 7.1, you need to update from glibc-2.2.4 to 2.2.4-19 or later. For details, see http://www.redhat.com/support/ .

2.For SuSE it is recommended to limit initial stack size, as a workaround for the Java virtual machine bug described by Sun (see http://java.sun.com/j2se/1.3/relnotes.html). Sun's workaround is: "Use `ulimit -2 2048` in bash shell or `limit stacksize 2048` in tcsh to limit the initial thread stack to 2 MB." Note that this is set automatically by Together.sh if you have JDK 1.3.1 installed, so you do not need to do it manually.

JVM information for LINUX

When running under LINUX, Together requires Sun Java2 (formerly JDK) to support some development features. J2SDK 1.3 is included in the Together installation for LINUX, and Together runs under its JRE.

Installing on LINUX

Together installation is carried out using the InstallAnywhere utility. The name of the installation file is `Together_nnnn_linux.bin`, where *nnnn* is a build number.

To install Together under LINUX:

1. Set executable permission rights for the installation file:

   ```
   chmod +x Together_nnnn_linux.bin
   ```

2. Run the installation file:

   ```
   ./Together_nnnn_linux.bin
   ```

3. Follow the on-screen instructions to complete the installation.

Starting Together on LINUX

If you have a floating license, you will need to install the Together floating license server on a remote network host accessible to the computer running Together. For complete instructions, see Chapter 4, "Using the Together License Server". Once license management has been configured, you can start Together as described below.

Launcher

The launcher file `$TG_HOME$/bin/Together.sh` is written to disk during installation. It sets appropriate environment variables including JAVA_HOME (required), and launches Together. You can edit the file to point to a different Java runtime location if you wish. You may have to edit the startup scripts in this file depending on the location to which you install it.

To start Together under LINUX:

1. If you have a floating license, make sure that the Together floating license server is properly configured and running.
2. Run `$TG_HOME$/bin/Together.sh`. If necessary for your license, this launcher automatically starts the floating license client installed with Together.

HP-UX systems

This section contains system requirements, installation, and start-up information for getting started using Together on systems running the HP-UX operating system from Hewlett-Packard.

System requirements

Table 5 lists the system requirements for using Together with HP-UX.

Table 5 System Requirements for HP-UX

	RESOURCE	REQUIREMENTS	NOTES
Hardware	CPU	9000/785 400 Mhz or faster	
	Memory	1 GB	Together has been tested on hardware with this configuration. Less memory may be acceptable but this has not been confirmed for the current release.
	Disk Space	150 MB	
	Video	Any compatible. 1024x768 minimum screen resolution	Recommended resolution: 1280x1024 or more
	CD Drive	Optional	Necessary only if installing from CD
Software	OS	HP-UX B.11.00 C	
	Java SDK	Sun® Java2™ SDK version 1.3	Together runs under this JRE
	License Manager	Together floating license client (built-in) and valid node-locked or floating license	Running license server on remote license host with valid license is required for floating licenses.
	Web Browser	Netscape Navigator 4.6 (or later)	Needed for viewing API documentation
Other	An internet connection is useful for obtaining software updates, node-locked licenses, and support, and for accessing TogetherSoft on-line resources.		

JVM information for HP-UX

You need to have a Java Virtual Machine (JVM) already installed on your system in order to run the installation program. If you do not have one, it is recommended to

install J2SDK 1.3 before installing Together. During Together installation, you will be asked to specify the Java VM on your system that will be used by Together.

Installing on HP-UX

Together installation is carried out using the InstallAnywhere utility. The name of the installation file is `Together_nnnn_others.bin`, where *nnnn* is a build number.

To install Together under HP-UX:

1. Set executable permission rights for the installation file:

   ```
   chmod +x Together_nnnn_others.bin
   ```

2. Set the PATH variable by adding the fully qualified path to the JVM's bin directory. For example:

   ```
   PATH=/usr/local/jdk1.3.0_03/bin:.:$PATH export PATH
   ```

3. Run the installation file:

   ```
   ./Together_nnnn_others.bin
   ```

4. Follow the on-screen instructions to complete the installation.

Starting Together on HP-UX

If you have a floating license, you will need to install the Together floating license server on a remote network host accessible to the computer running Together. For complete instructions, see Chapter 4, "Using the Together License Server". Once license management has been configured, you can start Together as described below.

Launcher

The launcher file `$TG_HOME$/bin/Together.sh` is written to disk during installation. You may have to edit the startup scripts in this file depending on the location to which you install it. You need to correctly define the JAVA_HOME environment variable so that it points to the directory where the Java runtime is installed on your system.

To start Together under HP-UX:

1. If you have a floating license, make sure that the Together floating license server is properly configured and running.
2. Run **TG_HOME/bin/Together.sh**. If necessary for your license, this launcher automatically starts the floating license client installed with Together.

Compaq Tru64 UNIX systems

This section contains system requirements, installation, and start-up information for getting started using Together on systems running the Tru64 UNIX operating system from Compaq.

System requirements

Table 6 lists the system requirements for using Together with Tru64.

Table 6 System Requirements for Tru64

	RESOURCE	REQUIREMENTS	NOTES
Hardware	CPU	600-MHz Alpha 21264A CPU or faster	Together has been tested on hardware with this configuration. Slower processors may be acceptable but this has not been confirmed for the current release.
	Memory	256MB minimum	512MB recommended
	Disk Space	200 MB	Required for Together and JDK installation
	Video	4MB tested, 8MB recommended 1024x768 minimum screen resolution	Recommended screen resolution: 1280x1024 or more
	CD Drive	Optional	Necessary only if installing from CD
Software	OS	Compaq Tru64 UNIX, V5.1	As tested
	Java SDK	Sun® Java2™ SDK version 1.3 or later	Together runs under this JRE
	License Manager	Together floating license client (built-in) and valid node-locked or floating license	Running license server on remote license host with valid license is required for floating licenses.
	Web Browser	Netscape Navigator 4.6 (or later)	Needed for viewing API documentation
Other	An internet connection is useful for obtaining software updates, node-locked licenses, and support, and for accessing TogetherSoft on-line resources.		

JVM information for Tru64 UNIX

You need to have a Java Virtual Machine (JVM) already installed on your system in order to run the installation program. If you do not have one, it is recommended to install J2SDK 1.3 before installing Together. During Together installation, you will be asked to specify the Java VM on your system that will be used by Together.

Installing on Tru64 UNIX

Together installation is carried out using the InstallAnywhere utility. The name of the installation file is `Together_nnnn_others.bin`, where *nnnn* is a build number.

To install Together under Tru64 UNIX:

1. Set executable permission rights for the installation file:

   ```
   chmod +x Together_nnnn_others.bin
   ```

2. In PATH, add the fully qualified path to the JVM.
3. Run the installation file:

   ```
   ./Together_nnnn_others.bin
   ```

4. Follow the on-screen instructions to complete the installation.

Starting Together on Tru64 UNIX

If you have a floating license, you will need to install the Together floating license server on a remote network host accessible to the computer running Together. For complete instructions, see Chapter 4, “Using the Together License Server”. Once license management has been configured, you can start Together as described below.

Launcher

The launcher file `$TG_HOME$/bin/Together.sh` is written to disk during installation. You may have to edit the startup scripts in this file depending on the location to which you install it. You need to correctly define the JAVA_HOME environment variable so that it points to the directory where the Java runtime is installed on your system.

To start Together under Tru64 UNIX:

1. If you have a floating license, make sure that the Together floating license server is properly configured and running.
2. Run **TG_HOME/bin/Together.sh**. If necessary for your license, this launcher automatically starts the floating license client installed with Together.

Other UNIX systems

Theoretically, Together should run on any operating system for which there is a Java Virtual Machine, and its development features should work on any platform for which there is a Java SDK. However, TogetherSoft limits development, testing, and support to the specific platforms mentioned in this guide. If you wish to try using Together on other UNIX variants with different JVMs, you can use LINUX system requirements as a basic guideline. Please understand that *you do so at your own risk*. TogetherSoft welcomes any feedback on results of running Together on platforms not currently supported.

Starting Together from the command line

You can launch Together from the command line of your operating system. This method is optional on some platforms (Windows, for example) and preferred on others. Using command-line startup enables you to use command-line syntax options to:

- Open a specific project on startup.

- Invoke Together with a shared remote configuration.
- Execute a specific module.
- Run Together in *command mode* (that is, without dialogs and user interactions).

You can use command-line syntax in batch files (or command, shell script, or other files) that you write and use to run Together. You can create such files and use them to run Together under different supported JVMs, with different classpaths, and so on.

Running Together in command mode supports the absence of user interactions (message boxes, dialog boxes, input requests) as much as possible. Command mode is useful for running Together modules or accessing other features via the API as part of some external automated process. For example, during a nightly build process you could invoke Together's HTML doc-generation module.

Command mode execution requires careful attention to the use of parameters to construct a proper command line. Check for the existence and accessibility of the necessary project and configuration files. When executing a module in command mode, Together automatically exits after completion.

Together system and other modules often display the interactive Save As dialog box prompting an output location and/or file. Check for this behavior by running the module from the Modules tab of the Explorer while running Together in normal mode. To avoid user interaction, you may have to alter a module (or write a new one) to automatically specify output parameters. Some modules have a switch that you can use to specify the output location when running in command mode.

CHAPTER

2

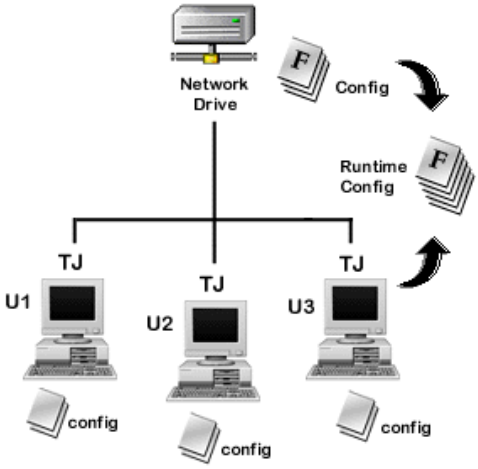
CONFIGURING TOGETHER FOR MULTIPLE USERS

This chapter includes the following topics:

Overview of a shared multi-user configuration

Together uses a tiered, hierarchical approach to configuring properties, meaning that as the Together administrator, you can configure options at three pre-defined levels: Default, Project, and Diagram. This enables you to provide a set of configuration properties (such as code-generation rules) for all users of the installation. You can specify which options users can not override, and which options users can configure with their own settings.

Figure 1 Together running on individual workstations



Setting-up multiple workstation installations

Multiple users with their own copies of Together running on separate workstations can access a shared multi-level configuration. As the Together administrator, you can store configuration properties for the group in a centrally accessible location on the network. Users launch Together from a batch or command file, specifying the path to the central configuration using the `-config.path` command-line switch.

> **NOTE:** In addition to the predefined configuration levels of default, project, and diagram, you can add new configuration levels to Together that users can access. The instructions that follow include a reference to this option.

To set up a multi-workstation shared configuration for a shared installation, complete these steps:

1. Install Together on your (the Together administrator's) workstation. Make a back-up copy of the original `./config` directory of the Together installation.
2. Run Together on your workstation.
3. If additional configuration levels are needed, create these as described in "Adding new configuration levels to predefined levels" on page 26.
4. Edit configuration options as required by choosing **Tools** | **Options |** ***applicable_level***, and clicking the **Levels** >> button. Mark individual options as "final" at the desired level for those settings that you do not want individual users to override from a lower level.

   See Chapter 3, "Configuring Options" for detailed instructions for setting-up options.
5. Copy the modified `./config` directory (plus any additional directories created for added config levels) to a shared network location (for example, `<server>/tg_shared_config`) that is accessible to all Together users who use this configuration.

Creating the start-up pointer file

To centrally share the entire configuration, you need to create a pointer file to define the centralized location and override Together's hard-coded configuration locations. Once you create the pointer file and define its location(s), you need to reference the pointer file in the `-config.path` switch when you start Together from the command line (or from a batch or command file). The most convenient way to create the pointer file is to let Together generate a copy of its own defaults.

To generate a default pointer file:

1. Run the Together launcher for your OS and pass:

   `-debug.config.saveDefaultPath=<path to file>`

   in the command line. For example:

   ```
   Together.exe- debug.config.saveDefaultPath=c:\TogetherSoft/
   Together6.0\lib\pointer.config
   ```

2. Together starts up and writes the specified file to disk. You can then close Together.

The file name used in the example is just a suggestion. You can use any name that does not conflict with Together property file names. You can store the pointer file anywhere, but it is probably most convenient to keep it in the `./ lib` directory of each local installation. Immediately after you generate it, the `pointer.config` file will contain the following lines:

```
config.level.$internal.group = basic
config.level.$internal.0 = $TOGETHER_LIB$/internal.config
config.level.$internal.1 = $TOGETHER_LIB$/path.config
config.level.$default.group = session
config.level.$default.0 = $TOGETHER_CONFIG$/*.config
config.level.$default.write = $TOGETHER_CONFIG$/changes.config
config.level.$commandLine.group = basic
config.level.$project.group = project
config.level.$workspace.group = project
config.level.$workspace.0 = $PROJECT_DIR$/*.config
config.level.$workspace.write = $PROJECT_DIR$/$PROJECT_NAME$.tws
```

You need to edit the highlighted (in **bold**) lines to point to your shared configuration. Note that paths must reference mapped drives (for example, this path will not work: `\\ appserver\tg_shared_config\*.config`). So if your shared configuration is on `appserver` which is locally mapped to drive `s:`, then your edited lines should look like this (Windows style paths shown):

```
config.level.$default.0 = s:\tg_shared_config\*.config
config.level.$default.write = s:\tg_shared_config\changes.config
```

Launching via the command line and pointer file

You should create for each user, or instruct users to create on their workstations, a start-up batch or command file that launches Together locally from the command line using

the `-config.path` switch to specify the path to the pointer file. In Windows, the command would look something like this:

```
c:\Together\bin\Together.exe -config.path=c:\Together\lib\pointer.config
```

Adding new configuration levels to predefined levels

You can add up to three additional configuration levels to the pre-defined levels of Default, Project, and Diagram. New levels must be inserted above the installation-wide Default level in the Options dialog order.

For example, a Corporate level could be added to enforce certain configuration settings across the enterprise. You could define the File prologue option in Source Code options so that all generated source code files contain the corporate copyright. Marking this final at the Corporate level prevents changes from lower levels.

To add configuration levels, follow steps:

1. Copy the contents of the `$TOGETHER_HOME$/config` directory to another location to create a separate set of configuration properties for the new level. Possible locations include a shared network location (as described in the previous section) or a local directory in your Together installation (e.g. `config/ corporate`).
2. Create a `path.config` file in the `./lib` directory and point it to the directory you set up for the new levels. Together searches this file and loads additional configuration levels from the locations specified. If the location in the previous step is shared, be sure to perform this second step on all the local machines that need to share the configuration.

Adding new configuration directories to a shared location

If you are creating the new levels for a configuration that will be shared from a central location, you have to create a set of configuration properties files for each new configuration level.

To create configuration properties directories for a shared location:

1. Copy the `./config` directory of your Together installation to the shared location and rename it to `tg_shared_config` (or some other meaningful name).
2. For each configuration level you plan to add, copy the `./config` directory of your Together installation to a new subdirectory of `tg_shared_config`. Rename each new subdirectory with the name of the level it represents, such as `tg_shared_config/corporate.`

> **TIP:** Create the subdirectories under your local config directory, test that you have defined the new levels correctly, and then copy the config structure to the shared location and modify the levels from there.

Adding new properties directories to a local installation

If you are creating the new levels for a configuration on a local installation, you have to copy the `./config` directory of your Together installation to another location and rename it with the name of the level it represents such as `./config/corporate`.

For example, to create a Corporate configuration level, first copy everything in the config directory to some other directory, such as `$TOGETHER_HOME$/config/corporate`.

Creating the path.config file

Before you create the path.config file, make sure that it does *not* already exist. If you are setting up a shared configuration, you need to place a copy of this file in the Together installation of *each* user.

Use a text editor to create the file `$TOGETHER_HOME$/lib/path.config`.

Add the following lines to the file:

```
=== path.config ===
 optionsEditor.level.$corporate.name = Corporate
    optionsEditor.level.$corporate.visible = true
    config.level.$corporate.name = Corporate
    config.level.$corporate.visible = true
    config.level.$corporate.0 = <path to this level>
    === path.config ===
```

Replace `<path to this level>` with the path to the properties files for the level you are defining, such as `$TOGETHER_CONFIG/corporate/*.config`.

> **NOTE:** `config.level.<level name>.<number> = <path>` defines the source for config files. For shared locations, use the full path. For local paths, specify the full path if outside your Together installation; within that structure you can substitute an appropriate Together System Macro (such as `$TOGETHER_LIB$, $TOGETHER_CONFIG$, $TOGETHER_HOME$`) as part of the path specification.

If you are adding more than one level, you should add some additional lines for each one. For example, to add a Division level:

```
...
optionsEditor.level.$division.name = Division
optionsEditor.level.$division.visible = true
config.level.$division.name = Division
config.level.$division.visible = true
config.level.$division.1 = $TOGETHER_CONFIG$/division /*.config
```

Viewing added configuration levels

To verify that you have correctly added a new level, run Together, open the Options dialog from the main menu, and click the Levels button. (If you created the new levels in a shared location, you will need to launch Together from the command line specifying the path to the configuration.)

In the case of our Corporate example, you should see the new Corporate level in the drop-down list of available levels.

In this way you can configure local installations to include company-wide configurations. Company-wide levels will be configured through `path.config`, and the installation-wide Default level will become user-specific.

Modifying default configuration levels

To get a server installation with user-specific configuration, the entire configuration order should be redefined. Together has the command line parameter

`-config.path=<path to file which defines all levels>.`

This option overrides hard-coded levels configuration. To get the hard-coded settings written to a file, pass

`-debug.config.saveDefaultPath=<path to file>`

in the command line. You will get a file, which can be used in the `-config.path` option to set the default behavior. This file can be modified to add new levels.

To make a new level visible in the Options dialog, you should include additional lines in the config file (such as misc.config):

```
optionsEditor.level.$corporate.name = Corporate
optionsEditor.level.$corporate.visible = true
optionsEditor.level.$department.name = Department
optionsEditor.level.$department.visible = true
```

`[...].name` is optional. If absent, the internal name will be shown (“corporate”).

`[...].visible` is necessary, because the level is not visible by default.

CHAPTER

3

CONFIGURING OPTIONS

This chapter explains how to configure options for an installation of Together. As the Together administrator, you can configure options on a global level, impacting all users.

This chapter includes the following topics:

Activating and deactivating modules

Some Together features, such as the GUI builder and testing framework, need to be activated before you can use them. This also applies to most integrations for third-party products such as Versant and Persistence Powertier.

> **NOTE:** By activating a module, you can also view options specific to the module. Similarly, deactivating a module removes the options specific to the module from the user interface.

To activate (or deactivate) a module, follow these steps:

1. From the Tools menu, choose Activate/Deactivate Features
2. Select the **Together Features** tab (or the **Integrations** tab)
3. Check the modules that you want to activate; uncheck the modules that you do not need to use. For your convenience, each module's path name and size is provided.
4. Click **OK**

Depending on the number of modules you have activated, it make take several seconds for Together to refresh the user interface.

Overview of configuration levels

You can set options for each of the three levels of Together's multilevel configuration:

- **Default level**: Settings apply to the entire Together installation, unless overridden at the project or diagram levels
- **Project level**: Settings apply only when a project is open, unless overridden at the diagram level
- **Diagram level**: Settings apply to a specific diagram (specifically, the diagram that is open at the time the option is set)

For any of the three levels, you can set options using a dialog in either *default* or *advanced* mode. Default mode allows you to set options for a single configuration level. Advanced mode allows you to set options at multiple levels without re-invoking the dialog.

Together allows you to apply settings for most of the options at any level. Consequently, options cannot be overridden at a lower (that is, more local) level.

When Together is installed as a server-based application, options are centralized and shared by all users. As the Together administrator, you can customize the configuration so that specific settings apply across the enterprise (Default level marked final), or globally for a team (Project level marked final). See Chapter 2, “Configuring Together for Multiple Users” for more information.

Setting options in advanced mode

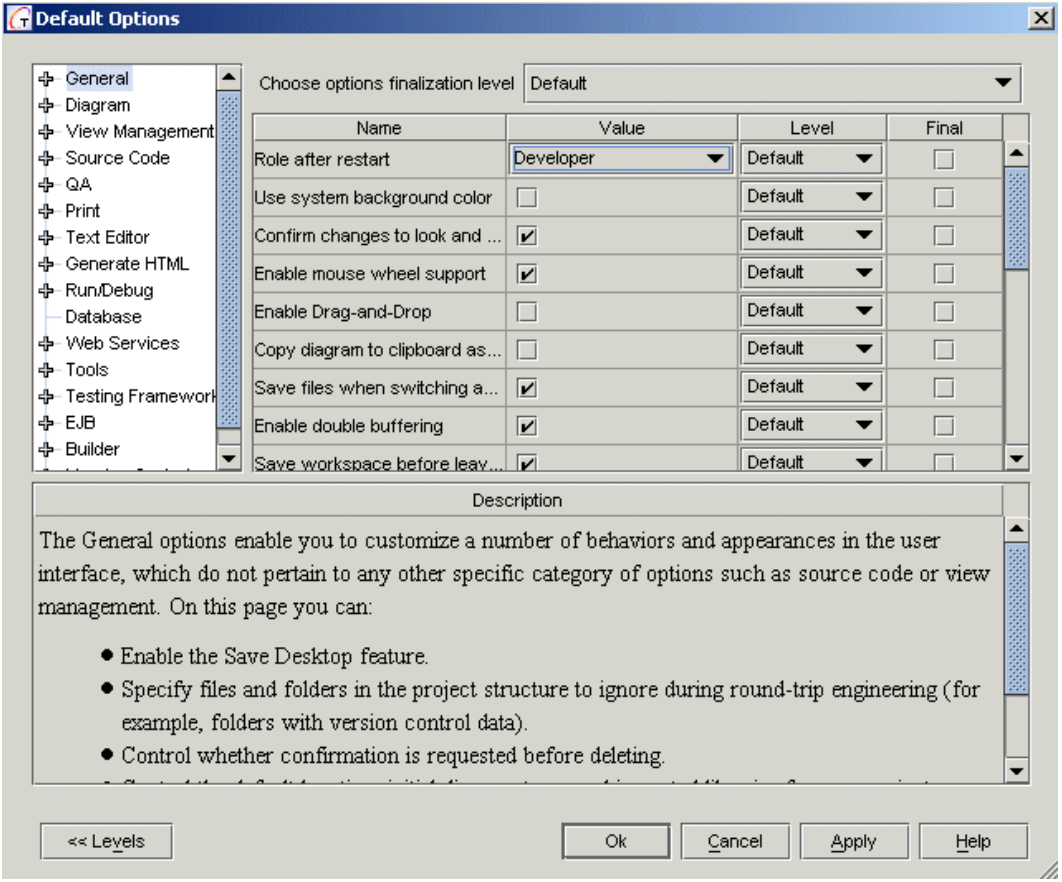
You can use the options dialog in advanced mode to set options at multiple levels without re-invoking the dialog. The options dialog for each level (default, project, diagram) includes a *Levels* button that you can use to toggle between default and advanced modes.

In contrast to default mode, advanced mode includes the columns *Level* and *Final*, as shown in Figure 2. The Level column allows you to select the configuration level. In this case, Level indicates the level at which each configuration option is currently set, assuming the default level definitions:

Default = Option is set at the Default level
Project = Option is set at the Project level
Diagram = Option is set at the Diagram level

The Final column indicates that an option cannot be overridden at a level more local than the one shown in the Level column. You can change any settings that are not marked final at a “higher” (that is, more global) level. In a local installation, you typically have complete control over your configuration and the settings at all levels. In a shared installation, you can change only those settings not marked as final by the system administrator in the shared configuration.

Figure 2 The Default Options dialog in advanced mode



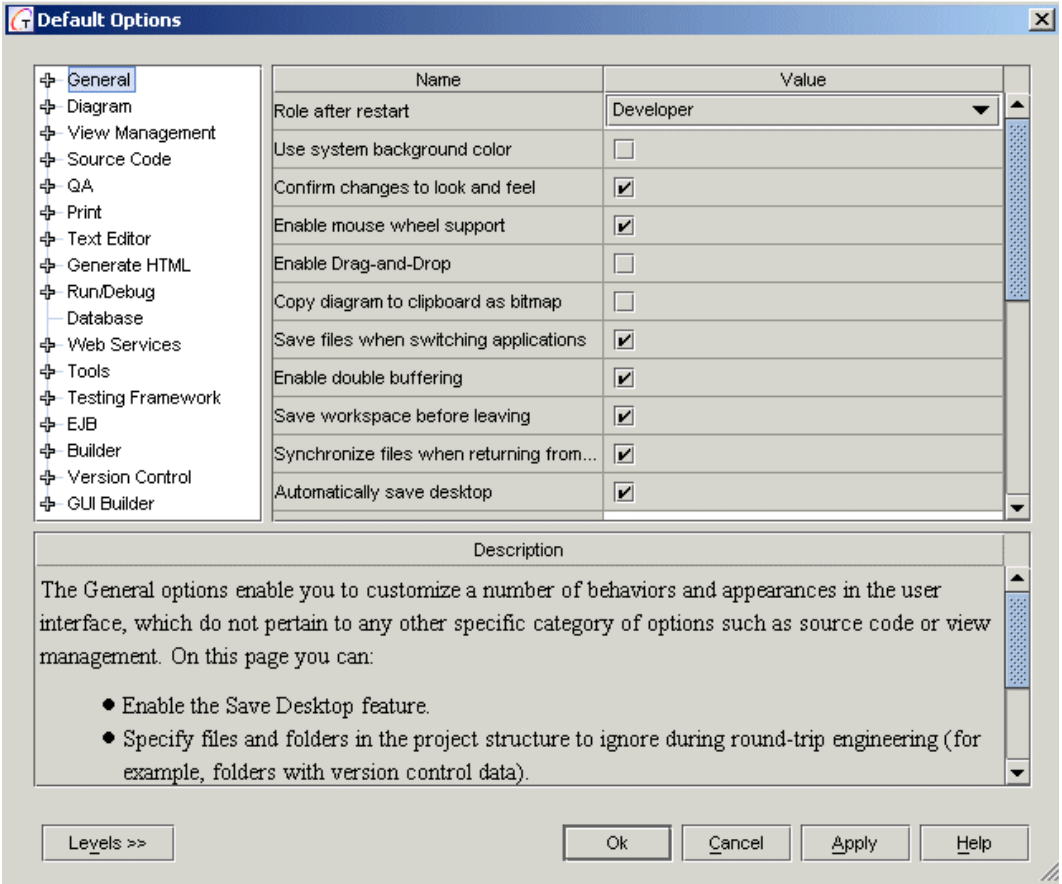
To set options in advanced mode, follow these steps:

1. Open a project. If you want to set options for a specific diagram, select the diagram.
2. Open the Options dialog for the lowest available level by choosing **Tools** | **Options** | ***available_level***.
3. Click the **Levels**>> button to enter advanced mode.
4. Set options for each level. You can often set options for a feature at *each* hierarchical level of the tree view. Select a node, regardless of its hierarchical level, to view the values that you can set.
5. Click **Apply** to save changes as you work.
6. Click **OK** to save changes and close the dialog.

Reference guide to options

This section summarizes the options you can configure in Together. Options are listed and grouped within the tree of the Default Options dialog (as well as the Project Options and Diagram Options dialogs), as shown in Figure 3. Help text for the nodes and the individual options appears in the Description field of the options dialog.

Figure 3 Tree listing options and option groups



Keep in mind that you can set the options at any one of three pre-defined configuration levels. Be sure you know what level you are working on before you change any option settings.

Default and project level options

Table 7 lists the options that you can set at the default and project levels. The table lists the options in the order they appear in the Default Options dialog.

> **NOTE:** If the user interface does not show the options listed for a specific feature, make sure that the feature is activated. Some features such as the testing framework and GUI builder must be activated before you can use them. See "Activating and deactivating modules" on page 29 for more information.

Table 7 Default and project level options

Option	Allows you to...	On this node you can...
General	Customize a number of behaviors and appearances in the user interface.	• Control background color and font properties. • Define your workspace role. • Set to auto-synchronize files when returning from other applications (such as an external editor or IDE). • Enable or disable the Saved Desktop feature that remembers your desktop settings between sessions. • Specify files and folders in the project structure to ignore during round-trip engineering (for example, folders with version control data). • Control the display of delete confirmations. • Control the default location, initial diagram type, and referenced libraries for new projects. • Control whether the Message pane opens on errors. • Enable or disable email exception reports to the Together development team. • Control whether Together will search archives for diagram files.
Diagram	Control a number of default behaviors and display properties that apply only to diagrams.	• Control the default link routing method. • Control the alignment, layout, justification, and initial maximum width of classes in diagrams. • Change the font used in diagrams. • Specify how association links are drawn and how and whether they are represented in classes (and therefore, indirectly, in source code). • Specify whether to generate metafile images of diagrams when saving diagrams. • Control the display of the background grid in the Diagram pane.

Table 7 Default and project level options (continued)

View Management	Control what you want to see and when. Specify how different kinds of elements display in the diagrams, or even whether they show up at all.	• Control the general level of detail shown in diagrams. • Control whether members in classes display in UML or Java format. • Show or hide subpackage contents in diagrams. • Control how JavaBean classes and C++ properties display in Class diagrams. • Show or hide referenced classes in diagrams. • Control the display of dependencies. • Show object class names and message numbers in Sequence diagrams. • Control the display of messages in Sequence diagrams. • Set banned destinations. • Specify the maximum call stack depth for Sequence diagrams generated from operations. • Show or hide aggregations of diagram elements and EJB elements.
Source Code	Control a number of default behaviors and appearances that apply to the formatting of source code during forward and reverse engineering operations.	• Specify the relative position of Attribute declarations and Operation declarations within Class declarations (Attributes first or Operations first). • Control how link attributes are handled when the destination is deleted. • Specify when code should be reformatted. • Specify the type of line separator for your OS. • Specify language-specific code formatting, optimization, name referencing, and import statement options. • Specify version-specific IDL formatting options such as indenting and format of comments. • Specify exactly how source code and comments are formatted (in-line breaks, space preservation, separators, and so on). Includes Javadoc comment formatting options for comments. • Customize source file prologue and epilogue text (the “Generated by” text at the head of source code files) for Java, C++, and IDL.
QA	Customize QA options. **Note:** To view this option, you need to activate the QA module.	• Specify the path to saved sets of audits and metrics. • Define the applicable scope of Quality Assurance.

Table 7 Default and project level options (continued)

Print	Set a number of defaults that apply to printing diagrams, files, and generated documentation.	• Set default paper size or define a custom one (for example, for printing on a plotter). • Set page orientation and margin sizes. • Set a number of other print options such as print zoom level, page border/footer, and so on.
Text Editor	In the Text Editor options, you can control a number of default behaviors and appearances that apply to the display of text in the Editor pane. **Note:** Source code formatting is not customized on this node. Use the Source Code options.	• Define the number of spaces inserted in text when you use the Tab key. • Define the font size. • Define the text color and style for code comments. • Define the text style for programming language reserved words. • Define the orientation of the cursor. • Customize the keyboard shortcuts for the editor. • Customize the way the editor works with specific kinds of files in a number of programming languages.
Generate HTML	Control the inclusion/exclusion of various content in the output of the standard HTML documentation generation facility (Tools | Generate HTML).	• Include or exclude author and version tags in generated output. • Specify all Javadoc settings. • Specify which visibility levels of classes to include in generated HTML output. **Tip:** You can set these options as you begin the documentation generation process. From the Generate HTML dialog, click Options to display the Options dialog with only the Generate HTML node visible. If you change any settings, they are used for the current documentation generation. operation only and then discarded. Your options settings for your configuration are not changed.
Run/Debug	Customize your runner/debugger.	• Define the root directory of the JDK for running and debugging applications from Together. **NOTE:** The default compiler is the Sun SDK, which is specified in the field JDK Home. By default it points to TGH/jdk. If the JDK is not a part of the Together installation, you should specify the correct folder. Otherwise compile/make/run commands will not work. • Define the location of the sources and current working directory. • Specify the project's run configurations. • Specify settings for the debugger. • Specify settings for JSP.

Table 7 Default and project level options (continued)

Database	Customize a number of common database properties that are used in operations with databases.	• Define the amount of time that Together will wait for a connection with DBMS. • Control whether Together replaces tables after generating a DDL. • Specify whether to use quoting symbols for identifiers in the resulting DDL.
Web Services	Set the application server that is used by default to register web services. Together can work in conjunction with other file-based development tools, such as compilers, debuggers, IDE's, and editors. The Tools node contains the external tool definitions for your configuration. The External Editor tool definitions are already pre-defined for you. Shell definitions for other tools are included, which you can use to set up interaction with an IDE or other tools. You can edit the fields to point these definitions to appropriate tools on your system. Menu commands for launching or interacting with the various tools are displayed on appropriate menus in the menu system. You can specify which menus should display the tool command in the Show in menu subnode. If you find you need more tool definitions beyond the user-defined, you can add more options to the `Tools` node. This is an advanced customization that involves editing the `tool.config` file.	
Tools	Edit various fields to point to appropriate file-based development tools (such as compilers, debuggers, IDEs, and editors). The Tools options contain the tool definitions for your configuration. The External editor tool is pre-defined for you.	
Testing Framework	Customize testing framework options. **Note:** To view this option, you need to activate the Testing Framework module.	
EJB	Set J2EE defaults	• Select the application server for J2EE deployment, which also determines which EJB specification new EJBs will be designed to meet. • Edit EJB suffixes that are used for EJB recognition.

Table 7 Default and project level options (continued)

Builder	Set options for the Builder.	• Choose to performing compilation prior to run/debug. • Reflect the compiling process in the Status bar. • Control the format of compiler output. • Choose the target folder for the generated makefile. • Set compiler options. • Specify the maximum permissible number of compilation errors.
Version Control	Enable and set up version control integration. By default, everything is set up to integrate with CVS (which is installed with Together). If you use an SCC-compliant version control system, choose SCC in the Use option. **Note:** To use an SCC version control system, Together must be running under Windows (NT/98/95) and Coroutine classes must be installed (default auto-installed and configured by the Together installer for Windows).	• Enable or disable version control integration. • Specify default interactions such as automatically getting files from version control when opening a project. • Specify what type of version control integration to use (CVS or SCC). • Set properties for CVS LAN and CVS Client-Server.

Diagram level options

You can set the following options at the diagram level:

- Diagram
- View Management
- Print

As these options can also be set at the default and project levels, see Table 7 for a description of each.

CHAPTER

4

USING THE TOGETHER LICENSE SERVER

The tasks described in this chapter are provided for the site administrator responsible for setting-up and maintaining the license server. The information is specific to the Together floating license server and does *not* apply to node-locked licenses.

This chapter includes the following topics:

About the Together license server

The Together license server provides an online, graphical user interface that allows the site administrator to perform the following tasks:

- Control end-user access to specific TogetherSoft products
- Allow specified users to check out licenses to use TogetherSoft products offline
- Control the amount of time each off-line license can be checked out
- Create time-out periods for licenses that have not been in use for extended periods of time
- Verify and report the number of licenses in use at any one time on the network by feature, version, user, and hostid
- Create usage reports for the number of licenses used on a daily, weekly, and monthly basis

> **NOTE:** In order to provide licenses to users in the event that your primary license server is offline, TogetherSoft recommends setting up both a primary and backup license server. However, using a backup license server is optional.

Following are additional notes for the primary and backup servers:

- Unlike the primary license server, you can not configure the backup server
- All configuration information is sent to the backup server from the primary server in 20 minute intervals
- The primary server allows users to checkout extended licenses; the backup server does not support this function.
- The Together client can contact the backup server only when the primary server is unavailable

System requirements

Before installing and configuring the Together license server, you need to have the following on your system:

- An installation of Sun Java2 SDK 1.3 (formerly JDK) on your system. J2SDK 1.3 is automatically installed with Together, and is also available from http://java.sun.com This is the recommended Java Virtual Machine (JVM) for Together.

  NOTE: You must use version 1.3 of the SDK. At the time of this release, the license server does not support version 1.4.

- Cookies enabled in your browser (Netscape or Internet Explorer)

Installing the license server

This section provides instructions for downloading and installing the Together license server. The instructions apply to both the primary license server and the backup license server. Installing a backup server is optional, but recommended in the event that your primary server is offline. For more information about the primary and backup servers, see “About the Together license server” on page 38.

NOTE: If you plan to use a backup license server, use separate computers for the primary and backup license servers.

IMPORTANT: You must install Sun Java2 SDK 1.3 to use the license server. See “System requirements” on page 39 for more information.

Installing on Windows

Together uses InstallAnywhere to guide you through the installation of the license server. For more information, see “About the Together license server” on page 38.

To start the installer program, follow these steps:

1. Download the `TogetherFLS_xx.exe` file from the `Windows` directory. Double-click `TogetherFLS_xx.exe` to launch the installer program.

Table 8 lists the information required by the installer program. For your convenience, space is provided for you to record your entries. This information pertains to all supported platforms (Windows, UNIX, and Linux).

Table 8 Information required by the installer program (for all platforms)

Required information	Notes	Your entry
Whether or not you accept the terms of the license agreement	Click **Accept** to proceed	
Location for the installation	Accept the default directory by clicking **Next**	
Java VM for the license server to use	The installer program lists the Java VM's currently available on your system You must use SDK 1.3.	
Whether you are installing a primary server or a backup server	See "About the Together license server" on page 38 for more information	
For a primary server: • Primary Server Name • Primary Server Port • Backup Server Name • Backup Server Port	To avoid problems when setting-up the primary license server, assign a dedicated port number to the primary license server	
For a backup server: • Backup Server Name • Backup Server Port	To avoid problems when setting-up the backup license server, assign a dedicated port number to the backup license server	
User name and password		
Whether or not you want to register the license server as a Windows service		

Installing on UNIX

Together uses InstallAnywhere to guide you through the installation of the license server. For more information, see "About the Together license server" on page 38.

To start the installer program, follow these steps:

1. Download the `TogetherFLS_xx.bin` file from the `Unix` directory.Set executable permission rights for the installer program:

   ```
   chmod +x TogetherFLS_xx.bin
   ```

2. Run the installation file:

   ```
   ./TogetherFLS_xx.bin
   ```

For a list of the information required by the installer program, see Table 8 on page 40. After completing the installation, follow the instructions in "Setting-up the license server" on page 41.

Installing on Linux

Together uses InstallAnywhere to guide you through the installation of the license server. For more information, see “About the Together license server” on page 38.

To start the installer program, follow these steps:

1. Download the `TogetherFLS_xx.bin` file from the `linux` directory.
2. Set executable permission rights for the installer program:
   ```
   chmod +x TogetherFLS_xx.bin
   ```
3. Run the installation file:
   ```
   ./TogetherFLS_xx.bin
   ```

For a list of the information required by the installer program, see Table 8 on page 40. After completing the installation, follow the instructions in “Setting-up the license server” on page 41.

Setting-up the license server

After installing the license server, you need to complete *each* of the tasks described in this section to set up the server:

1. Starting the backup server (if using a backup server)
2. Starting the primary server
3. Uploading a license

Starting the backup license server

These instructions apply only if you installed a backup license server. The backup license server does not require any set-up tasks. However, if you decide to use a backup license server, you need to start the backup server in order to setup the primary license server.

To startup the backup license server, complete the appropriate step based on your operating system:

- For **Windows**: From the Start menu, select **Programs** | **TogetherFLS_xx | startup.bat**
- For **UNIX**:
  1. Change directories as follows:
     ```
     cd INSTALL_DIR/TogetherFLS_xx/jakarta-tomcat-4.0.1/bin
     ```
  2. Run `startup.sh`

  If you are unable to run the startup script (`startup.sh`), check the following:

  - Verify that proper permissions have been set
  - Verify that the backup license server was assigned a dedicated port number during installation (that is, no other servers are using the port number)

Starting the primary license server

If you are using a backup license server, you need to start the backup server as described in the previous section before starting your primary license server.

To set up and start the license server, follow these steps:

1. For **Windows**: Select **Programs** | **TogetherFLS_xx | startup.bat** from the Start menu;

   Or, for **UNIX**:

   1. Change directories as follows:
      `cd INSTALL_DIR/TogetherFLS_xx/jakarta-tomcat-4.0.1/bin`
   2. Run **startup.sh**

   If you are unable to run the startup script (`startup.sh`), check the following:

   - Verify that proper permissions have been set
   - Verify that the primary license server was assigned a dedicated port number during installation (that is, no other servers are using the port number)
2. Open a browser and enter the url of your primary host and primary port number: `http:\\PRIMARY_HOST:PRIMARY_PORT\`
3. Follow the instructions for “Uploading a license” on page 42

Uploading a license

Before you can accomplish any administrative tasks, you must upload a valid license to the system.

To upload a license, follow these steps:

1. Log into the system.
2. Click **Manage Floating Licenses** on the main menu.
3. Click **Browse** to locate the license to be uploaded. The name of the file containing the license has the extension `.float`.
4. Click **Add License**. The license server displays information for the license, as follows:
   - **Serial Number:** A unique number that identifies the license.
   - **Customer Name:** The contact name of the user as generated in the license.
   - **License Type:** The TogetherSoft product assigned to the license.
   - **Total Instances:** Total number of users who can use Together at one time.
   - **Max Checkout Duration (hours):** The total number of hours a license instance can be checked out for offline use.
   - **Primary Host ID:** The host ID of the server that you assigned as the primary license server during installation.
   - **Backup Host ID:** The host ID of the server that you assigned as the backup license server during installation.

- **Available Instances:** Total number of available licenses (Total Instances less the number of current users)

If an error occurs, contact: support@togethersoft.com. You can expedite service by providing the following information:

- Your name
- Your company's name
- Your server's HostId
- Your license file (`*.float`)

Using the license server

Before completing the tasks explained in this section, you need to follow the instructions for:

- "Starting the backup license server" on page 41
- "Starting the primary license server" on page 42
- "Uploading a license" on page 42

Logging in to the administrative portal

To log into the portal, type the user name and password that you selected during the license server installation. Once logged into the system, you can log out by clicking the Log Out of System link, or by closing the browser.

Adding users

In order for an individual to use a license purchased by your company, you must first add them as a user to the system. The instructions in this section assume that you are logged in and have uploaded a valid license.

To add a user to the license server, follow these steps:

1. From the main menu, click **Add A User**.
2. Complete the required fields, and then click **Add** to submit your information. Note that *UserName* should match the user's network login.
3. Review the information and click **Add,** or press Back (on your browser) to edit.
4. Select the permission name and license type, and enter the maximum length of time (in hours) for an off-line license.
   - *Permission Name* is set to *Reserve* by default. This allows users to reserve both online and offline licenses.
   - *Maximum Duration* applies to offline licenses. Setting this field to zero (0) prevents the specified user from obtaining an offline license.
5. Click **Add**.

Finding users

Finding a user is an essential step to updating user information or generating custom user reports. The instructions in this section assume that you are logged in and have uploaded a valid license.

To find a user, follow these steps:

1. From the Main Features menu, select **Find A User**.
2. Enter information in any of the fields that you want to query
   - Entries are case sensitive
   - To denote a wildcard, use the percent character (%)
   - For a list of all users, enter **%** in a field
3. Click **Find**.
4. Select the user that meets your criteria and click **Select User**. The screen lists user information and permissions, and provides links for:
   - Updating user information
   - Updating floating license permissions
   - Viewing a list of licenses currently checked-out by the user
   - Checking-in the user's license

Updating user information

Prior to using the instructions in this section, follow the steps for "Finding users" on page 44.

To update a user, follow these steps:

1. After finding the appropriate user, select **Update Personal Information For *UserName***. Edit the information and then click **Update**.
2. Review the revised information, and click **Update** or press **Back** (on your browser) to reenter.
3. Update the user's permissions. You can add a new permission, delete an old permission, or update a current permission.

   NOTE: A user can have multiple permissions. To update a permission, first delete the existing permission and then add a new permission with the updated information.

Creating usage reports

The Together license server allows you to generate the reports listed in Table 9.

Table 9 Types of reports

Report	Description
List All Online Users	Lists all current users of the system
List All Checked Out Licenses	Lists all currently checked out licenses
List Usage History	Provides a list that indicates each time a license was reserved or checked out from the system during a specified date range
List Checkout History	Lists all checked out licenses
List Available Licenses	Lists all available licenses in the system

To create a usage report, follow these steps:

1. From the Main Features menu, click **Create License Usage Reports**
2. Select the type of report you would like to generate, as listed in Table 9, and then click **Generate Report**.

   NOTE: The List Usage History report requires that you specify a range of dates.

3. Once the report is generated, you can sort the columns by clicking on the column header.

Managing floating licenses

You can use the *Manage Floating Licenses* link under the Main Features menu to add (that is, upload) or remove floating licenses. Note that this is the license used by the license server, as opposed to the licenses that you assign to users.

To add a license, see “Uploading a license” on page 42. To remove a floating license, click **Remove** for the applicable license.

Checking-in licenses

When you check-in a user’s license, you end their ability to use the Together product associated with the license. Prior to using the instructions in this section, follow the steps for “Finding users” on page 44.

> **NOTE:** This feature applies to licenses that have been checked-out for offline use. Together automatically checks-in the license the next time the user starts Together while connected to the network.

To forcibly check in a user's current license, follow these steps:

1. After finding the appropriate user, select the link **Check In License(s) Used By *UserName***; A new window appears with the license currently checked out.

2. Click on the link of the license you want to check in.
3. Close the window.